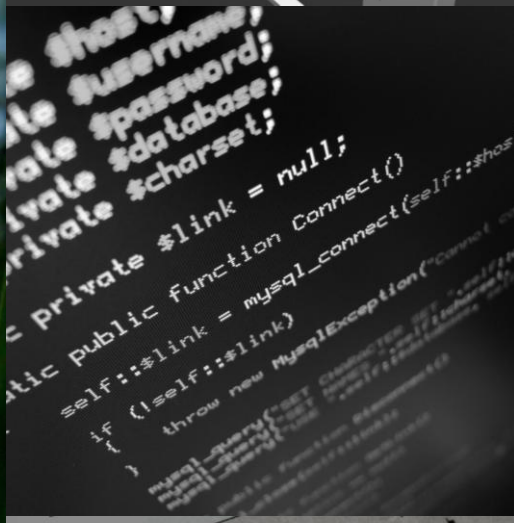
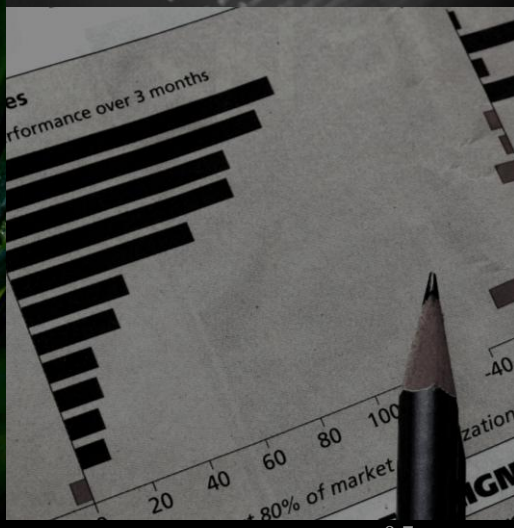




v0.7

企业软件保证成熟度模型

目录

简介

成熟度模型的背景信息

动机和基本原理

该软件保证计划的成熟度模型是根据一些尚在理论阶段的指导原则和目标构建的。

首先，要正确构建保证计划，组织就需要执行各种不同的措施，而改变组织的一贯行为很难。要在不增加间接成本的前提下高效地构建保证计划，必须在每个改进阶段明确定义短期目标，并逐步完善长期目标。

第二，必须有相应的标准来衡量保证计划的改进效果，且该衡量标准应简单易懂。若措施的定义含糊不清，会导致难以在正常情况下使用成熟度模型，并且使用时很容易出错。

第三，一个保证计划无法用于所有类型的组织。应该针对每个组织所面临的特定风险，根据软件的构建方式和业务用途来构建相应的保证计划。因此，成熟度模型应该根据业务类型来定义改进方案，并使用建议的路线图及案例分析来阐明常见用例。

角色定义

开发人员

负责软件的低级设计和实施的人员

架构师

负责高级设计和大型系统工程技术的人员

管理人员

负责日常管理开发人员的人员

QA 测试人员

负责软件的常规测试和发布前验证的人员

安全审核人员

具备软件技术安全知识的人员

企业所有者

负责对软件及其业务要求做出关键决策的人员

运营支持人员

负责提供客户支持或直接技术支持的人员

目标受众

本文档介绍了与软件生产中众多角色相关的信息，可供各类人员使用。采用以下建议可以简化您开始使用 SAMM 的过程

管理人员、企业所有者、安全审核人员、架构师

对于希望了解如何领导软件保证计划和管理计划进展的人员，应该先阅读“成熟度领域”一节中每页的概述，以了解高级安全功能。掌握概要信息对于之后的阅读非常有用，您随后可以仔细阅读与您的组织相关的路线图和案例研究。如果需要，您可以返回到“成熟度领域”一节以了解特定功能和目标的详细信息。

开发人员、架构师、QA 测试人员、安全审核人员、运营支持人员

若要了解如何在项目级别执行特定措施以改进安全功能的置备过程，可以选择阅读“成熟度领域”一节中介绍“目标”和“措施”的部分。如果需要，可以参阅相关的目标和其他资源，以了解背景和上下文信息。要了解功能将来的进一步改进，还应该查看后续目标。

使用成熟度模型

该成熟度模型介绍了组织用以降低安全风险并提高保障水平的各种措施。在最高级别，包含代表一般措施的以下四种**规范**：

在每个规范下，按照三种**功能**进行分组，其中每种功能分别代表组织能够逐步改进的安全相关措施的范围：

每种功能都有三个连续的**目标**，这三个目标为组织定义了针对指定功能的逐渐复杂的目标。依照本文档中的指导，组织可以实现一个目标。实现第一个目标之后，可以继续实施指定功能的下一个目标。一般而言，每个功能下的连续目标代表：

大多数组织已执行了其中的某些措施，因此，必须执行初步评估，以便根据已完成的目标为每个功能进行评分(0-3)，将组织的绩效与十二种功能关联起来。之后，组织应该根据业务驱动因素和特定于组织的信息重复执行这一过程，方法是选择要改进的功能，然后完成每个功能的下一个连续目标。

成熟度區域

此段落會提供「成熟度模型」本身四個組成區塊的定義。在四項高階「訓練課程」中，每一項訓練都含有三種安全「功能」，定義了可以執行的安全相關活動廣度。同樣地，每一個「功能」也都擁有三種佈建的層級以搭配相對應的活動、成功指標、人事額外負荷以及相關成本。

學習與指導方針

〈學習與指導方針〉的功能，是針對軟體生命週期的相關人員，給予應用程式安全的相關知識。專案團隊存取資訊的管道獲得改善後，就能夠更主動地識別並減緩組織中所發生的特定安全風險。

在所有改善的目標中，其中一個重大主題是提供員工訓練，而實行方法則為透過講師講解課程，或是以電腦為基礎的模組進行。一個組織漸漸地完成這些層級後，就可以建立一個廣泛的訓練基礎，由開發人員開始並擴及組織裡的其他更多角色，員工可慢慢累積以角色為基礎的認證，確保對教材的理解。

此功能在訓練之外也需要將安全相關資訊運用到指導方針中，以做為員工的參考資訊。如此便可建立一個基礎，讓貴組織建立安全實作的基線預期，並在稍後採用指導方針後，以累加的方式進行改善。

	EG 1	EG 2	EG 3
目標	**為開發技術人員提供管道，以獲得與安全程式撰寫與部署相關的資源。**	**配合安全開發的角色特有指導方針，對軟體生命週期內的所有人員給予訓練。**	**規定完整的安全訓練，並對具備基線知識的人員給予認證。**
活動	進行技術安全警覺訓練 建立並維護技術指導方針	進行角色特有的應用程式安全訓練 利用安全指南加強專案團隊表現	建立應用程式安全支援入口網站 建立以角色為基礎的檢驗/認證
結果	提高了開發人員對於程式碼層級常見問題的警覺性 在具備基本安全最佳實作的狀況下維護軟體 為技術人員設定安全專業知識基線 啟用基線安全知識的質化安全檢查	加強點對點警覺性，注意可能導致產品、設計與程式碼層級安全弱點的問題 建立計劃以修補進行中專案的弱點與設計缺陷 在需求、設計以及開發階段啟用質化安全檢查點 更進一步瞭解安全問題，有助於更加主動的安全計劃	分別在進行中與舊版程式知識庫中，同時進行更有效率的弱點矯正 快速瞭解新的攻擊與威脅，並減緩風險 判斷技術人員對安全的專業知識，並對照一般標準進行衡量 建立對安全警覺的適當鼓勵措施

學習與指導方針

EG 1

為開發技術人員提供管道，以獲得與安全程式撰寫與部署相關的資源。

活動

進行技術安全警覺訓練

無論在組織內部或外包執行，對技術人員的安全訓練都必須包括應用程式安全的基本原則。一般來說，訓練的進行可以在 1 到 2 天的時間內，透過講師指導完成，或是透過以電腦為基礎搭配模組完成訓練，而每一位開發人員所需花費的時間量也差不多。

課程內容應包括概念性與技術性資訊。適當的課程主題包括有關輸入驗證、輸出編碼、錯誤處理、登錄、驗證、授權等的高層級最佳實作。最好也可以講解其他常見軟體弱點，例如 OWASP Top 10 的網路應用程式或是更適當的修改清單，瞭解為其他範例（內嵌裝置、客戶伺服器應用程式、後端交易系統等）所開發的軟體。請儘可能的在套用的特定程式撰寫語言中，使用程式碼範例與課堂練習題。

為了執行此訓練，我們建議貴組織制訂年度安全訓練，並根據開發人員數舉行所需的課程（以講師指導或以電腦為基礎的方式進行）。

建立並維護技術指導方針

請針對開發人員撰寫一份清單，列出核准的文件、網頁以及技術性備註，以提供技術特有的安全建議。這些參考文件的準備，可以從網際網路上許多公用的資源中取得。如果開發環境中充滿高度專業或專屬技術，可指派具備安全專業知識的資深員工長期制訂安全注意事項，以特定的方式建立上述知識庫。

請確定管理人員瞭解這些資源，並對新進員工簡介資源預期的使用方式。請嘗試保持參考清單的簡短並隨時更新，避免雜亂與不相關性。只要清單達到舒適好讀的程度，這份技術參考清單就可做為質化檢查清單，確定員工閱讀並瞭解指導方針，而且在開發程序中遵守內容。

結果

- 提高了開發人員對於程式碼層級常見問題的警覺性
- 在具備基本安全最佳實作的狀況下維護軟體
- 為技術人員設定安全專業知識基線
- 啟用基線安全知識的質化安全檢查

成功指標

- 過去 1 年中，有超過 50%的開發人員接受過安全問題的簡報
- 過去 1 年中，有超過 75%的資深開發/架構人員接受過安全問題的簡報
- 在初次訓練的三個月內實施技術性指導方針

成本

- 訓練課程建置或授權
- 持續維護技術指導方針

人事

- 開發人員(1 至 2 天/年)
- 架構人員(1 至 2 天/年)

相關目標

- 〈標準與法規遵從〉-2
- 〈安全需求〉-1
- 〈防禦設計〉-1

相關法規遵從

- PCI v1.1，第 6.5 節
- PCI v1.1，第 12.5.1 節
- PCI v1.1，第 12.6 節

EG 2

學習與指導方針

配合安全開發的角色特有指導方針，對軟體生命週期內的所有人員給予訓練。

結果

- 加強點對點警覺性，注意可能導致產品、設計與程式碼層級安全弱點的問題
- 建立計劃以修補進行中專案的弱點與設計缺陷
- 在需求、設計以及開發階段啟用質化安全檢查點
- 更進一步瞭解安全問題，有助於更加主動的安全計劃

其他成功指標

- 過去 1 年中，有超過 60%的開發人員接受過訓練
- 過去 1 年中，有超過 50%的管理/分析人員接受過訓練
- 過去 1 年中，有超過 >80%的資深開發/架構人員接受過訓練
- 超過 3.0 的李克特量表得分，認為訓練課程有其效用

其他成本

- 訓練資源庫建置或授權
- 具備專業知識的員工親自教導

其他人事

- 開發人員(2 天/年)
- 架構人員(2 天/年)
- 管理人員(1 至 2 天/年)
- 業主(1 至 2 天/年)
- QA 測試人員(1 至 2 天/年)
- 安全稽核人員(1 至 2 天/年)

相關目標

- 〈弱點管理〉-1
- 〈架構檢閱〉-2
- 〈防禦設計〉-2

相關法規遵從

- PCI v1.1，第 6.5 節
- PCI v1.1，第 12.5.1 節
- PCI v1.1，第 12.6 節
- PCI v1.1，第 12.9.4 節

活動

進行角色特有的應用程式安全訓練

對於其職能的角色內容著重於應用程式安全的員工，提供安全訓練。一般而言，訓練的進行可透過講師指導，在 1 到 2 天的時間內完成，或是透過電腦式訓練模組完成，每個人所需花費的時間量差不多。

對於管理人員與需求規劃人員，課程內容應包含安全需求計劃、弱點與意外管理、威脅模型分析以及誤用/濫用案例設計等主題。

測試人員與稽核人員的訓練應著重於訓練員工瞭解並更有效率的分析軟體，檢查是否有安全相關的問題。課程本身也應該包括程式碼檢閱、架構與設計分析、執行階段分析以及有效安全測試計劃等技術。

將目標開發人員與架構人員的技術訓練課程擴展至其他相關主題，例如：安全設計模式、工具特有的訓練、威脅模型分析以及軟體評估技術。

為了執行此訓練，建議您訂出年度安全警覺訓練與週期性專業主題訓練。課程的頻率可根據每個角色的人數決定（可以講師指導或以電腦為基礎的方式進行）。

利用安全指南加強專案團隊表現

不論是使用組織內部或外部的專家，請確定專案團隊配有可供諮詢的安全專精員工。不僅如此，這一項指導資源應該在組織內部通告，以確定員工都知道可取用此資源。

提供指導的員工可以延攬組織內有經驗的個人組成指導團，並利用特定比例的上班時間（最多為 10%）執行指導活動。指導人員應該與每個人建立良好的溝通，以確保大家互相瞭解彼此專業，以便有效率的提報問題。

雖然指導人員可以在軟體生命週期中的任一時間點提供諮詢，但比較適當的時機為：產品概念化初始階段、功能或詳細設計規格完成之前、開發過程與測試計劃中產生問題，以及作業安全意外發生時。

以長時間來看，組織內部的指導資源網路可以做為整個組織內的連絡點，以溝通安全相關的資訊，且相較於純粹的集中式安全團隊，他們對於進行中專案的熟悉程度較佳，因此也可當作當地資源。

學習與指導方針

EG 3

規劃完整的安全訓練，並對具備基線知識的人員給予認證。

活動

建立應用程式安全支援入口網站

建立網站所根據的書面資源，主題均與應用程式安全、建立與通告集中式儲存區（通常用於內部網站）相關。組織可使用自身認為合理的方式建立指導方針，但首先必須建立核准委員會與直觀的變更控制程序。

除了最佳實作清單、工具特有的指南、FAQ 以及其他文章等形式的靜態內容外，支援入口網站也應該有互動性的元件，例如：郵寄清單、網路論壇、或是 Wiki 功能，以讓組織內部資源可以針對安全相關主題進行交流，並將資訊整理起來當作未來參考。

內容應根據平台、平台、程式語言、在特定第三方資源庫或骨架上的適用性、生命週期階段等因素，加以分類並讓員工得以輕易搜尋資訊。建立軟體的專案團隊應在產品開發過程中，及早自我調適以配合未來要遵循的特定指導方針。在產品評估過程中，應該使用適用的指導方針清單與產品相關的討論做為稽核標準。

建立以角色為基礎的檢驗/認證

您可依照角色或訓練課程/模組進行能力測驗的實施與管理，以測試學員對於課程的理解以及安全知識的運用。一般而言，應該根據各角色的課程為基準實施測驗，並將最低及格分數設定在大約 75%的正確度。而員工每年都必須接受適用的訓練或複習課程，至少每半年要參加一次認證測驗。

根據及格/不及格標準或是傑出表現，列出員工排名，如此一來，安全相關的活動可限制參加者必須由具備特定認證層級者簽核，才能完成活動，例如：未認證的開發人員在未取得已認證架構人員的明確核准之前，就無法實施其設計。這樣即可對各專案提供更細微的視角，以根據個人的責任追蹤安全決策。此做法可以就整體而言提供基礎，針對員工是否做出應用程式安全的相關完善商業決策，予以獎勵或懲罰。

結果

- 分別在進行中與舊版的程式知識庫中，同時進行更有效率的弱點矯正
- 迅速瞭解並減緩新的攻擊與威脅
- 判斷技術人員對安全的專業知識，並對照一般標準進行衡量
- 建立對安全警覺的適當鼓勵措施

其他成功指標

- 過去 1 年中，有超過 80%的員工取得認證

其他成本

- 認證測驗的建置或授權
- 持續進行應用程式安全支援入口網站的維護與變更控制
- 實施員工認證所產生的人力資源與額外負荷成本

其他人事

- 開發人員(1 天/年)
- 架構人員(1 天/年)
- 管理人員(1 天/年)
- 業主(1 天/年)
- QA 測試人員(1 天/年)
- 安全稽核人員(1 天/年)

相關目標

- 〈標準與法規遵從〉-2 和 3

相關法規遵從

- PCI v1.1，第 2.2 節
- PCI v1.1，第 6.5 節
- PCI v1.1，第 12.5.1 節
- PCI v1.1，第 12.6 節
- PCI v1.1，第 12.9.4 節

標準與法規遵從

〈標準與法規遵從〉的功能著重於瞭解與符合組織外的法規遵從需求，同時在顧及組織軟體專案的商業目的前提下，推動內部安全標準去配合這些需求。

此功能之中，推動改善的主題思想著重於專案層級稽核，此稽核動作會收集組織行為相關資訊，以檢查是否達到預期結果。 組織可循序導入例行性稽核，並將稽核的深入程度隨著時間漸漸增加，達到整體變更目的。

此功能的佈建十分精細，整個組織必須充分理解內部標準與外部法規遵從驅動因子，同時讓專案團隊保持低度檢查點延遲，以確保可察覺所有不符預期的專案運作情形。

目標	**瞭解組織相關的審查與法規遵從驅動因子。**	**建立安全與法規遵從基線，並瞭解每件專案的風險。**	**對組織整體的標準與政策進行法規遵從需求制訂與專案衡量。**
活動	識別並監控外部法規遵從驅動因子 建立與維護法規遵從檢查清單	建立安全與法規遵從的內部標準 建立專案稽核實作	建立專案的法規遵從標準 採用稽核資料收集的解決方案
結果	針對具有正面結果的第三方稽核處理提升保證性 根據法規遵從需求的優先順序調整內部資源 及時發現對貴組織有影響的修訂中法規需求	專案團隊對於安全與法規遵從預期的認知 業主可以更加瞭解產品線中的特定法規遵從風險 透過機會性安全改善，將有效達成法規遵從的方法進行最佳化	可從組織層級看見未遵循法規但卻被接受的風險 將專案層級的法規遵從保證性具象化 精確的追蹤以往的專案法規遵從歷史記錄 高效率的稽核程序可運用工具以減少人力支出
	SC 1	SC 2	SC 3

標準與法規遵從

SC 1

瞭解組織相關的審查與法規遵從驅動因素。

活動

識別並監控外部法規遵從驅動因子

一個組織可能必須遵從各式各樣的法規，然此活動特別針對某些法規，也就是對組織建立或使用軟體和/或資料的方式有著直接或間接影響者。若可行的話，請要求內部員工遵從法規。

根據組織的核心業務，研究並識別出必須遵從或被視為業界標準的第三方法規標準。可能的標準有：《美國企業改革法案》(Sarbanes-Oxley Act, SOX)、《支付卡產業安全標準》(Payment Card Industry Data Security Standards, PCI-DSS)、《醫療保險可攜性和問責性法案》(Health Insurance Portability and Accountability Act, HIPAA) 以及其他標準。在詳讀並瞭解所有第三方標準後，請收集與軟體及資料相關的特定需求，並建立一個整合清單，列出每一項驅動因子（第三方標準）與安全，及其對應的各特定需求。在此階段中，試著捨棄選用或建議使用的所有需求，盡量降低需求的數量。降低需求的數量。

至少每半年進行一次研究，以確保組織掌握第三方標準變更的最新動態。此活動可以根據組織所屬的產業別與法規遵從的重要性，變動參與的資源與人力，但務必要確實執行此活動。

建立與維護法規遵從檢查清單

根據來自法規遵從驅動因子的軟體與資料相關需求整合清單，建立每項需求的相對回應指令，讓此清單更加詳盡完備。各回應（有時又稱為「控制指令」）應保有一種概念，也就是組織為確保符合需求（或是說明為何此需求不適用）所作的努力。

因為典型的稽核實作通常會檢查控制指令是否足夠，再對照控制指令衡量組織，所以精確的呈現實際組織的實作是非常重要的。另外，只要先建立簡單、少量又符合基線法規遵從的程序元素，就能符合許多需求，進而讓組織實際達成較佳保證性的目的。

從整合清單開始進行作業，辨識出重大差異，並在建立保證性方案相關事宜時，將辨識結果納入未來規劃考量。與組織的利害關係人溝通目前與法規遵從目標的差異，以確保利益關係人瞭解未遵從法規的風險。

請至少每半年一次向利害關係人更新並檢閱控制指令。根據法規遵從驅動因子的數量，亦可依需要增加更新的頻率。

結果

- 針對具有正面結果的第三方稽核處理提升保證性
- 根據法規遵從需求的優先順序調整內部資源
- 及時發現對貴組織有影響的修訂中法規需求

成功指標

- 在過去 6 個月內曾進行過超過 1 場法規遵從探索會議
- 在過去 6 個月內曾完成並更新法規遵從檢查清單
- 在過去 6 個月內曾與利害關係人進行過超過 1 場法規遵從檢閱會議

成本

- 法規遵從檢查清單的初始建立與持續維護

人事

- 架構人員(1 天/年)
- 管理人員(2 天/年)
- 業主(1 至 2 天/年)

相關目標

- 〈策略規劃〉-1

相關法規遵從

-

SC 2

標準與法規遵從

建立安全與法規遵從基線，並瞭解每件專案的風險。

結果

- 專案團隊對於安全與法規遵從預期的認知
- 業主可以更加瞭解產品線中的特定法規遵從風險
- 透過機會性安全改善，將有效達成法規遵從的方法進行最佳化

其他成功指標

- 過去 6 個月內，有超過 75%的員工曾接受內部標準的簡報
- 超過 80%的利害關係人瞭解對內部標準的法規遵從狀態

其他成本

- 內部標準的建置或授權
- 遵從內部標準與稽核相關法規帶來的各專案額外負荷

其他人事

- 架構人員(1 天/年)
- 管理人員(1 天/年)
- 安全性稽核人員(2 天/專案/年)

相關目標

- 〈學習與指導方針〉-1 和 3
- 〈策略規劃〉-2
- 〈安全需求〉-1 和 3
- 〈防禦設計〉-3
- 〈程式碼檢閱〉-3
- 〈架構檢閱〉-3
- 〈基礎架構強化〉-3

相關法規遵從

活動

建立安全與法規遵從的內部標準

請先使用目前的法規遵從檢查清單，檢閱法規標準並備註任何的選用或建議安全性需求。組織也應該進行小量研究，以發現相關法規遵從中潛在的未來變更。

根據已知的安全商業驅動因子，配合任何額外需求以擴展清單。最簡單的方法通常是查閱提供給開發人員的現有指導方針，並匯集成最佳實作集。

將常見/類似的需求分組，並將每個群組重寫為更普遍/簡化的指令，這些指令不但要符合所有的法規遵從驅動因子，也必須提供一些額外的安全價值。對每個群組進行此程序，以求建立可直接對應回法規遵從驅動因子與最佳實作的內部標準集。

統一後的標準，其中的需求不得讓專案團隊難以遵從或造成過多成本。而必須讓 80%的專案得以在最小的干擾情況下順利遵從標準。若要達到此目標，必須建立良好的溝通方案，以通告新的內部標準，並在需要時提供法規遵從方面的協助。

建立專案稽核實作

為專案團隊建立簡易的稽核程序，以讓團隊可以要求並接收對內部標準的稽核。稽核通常由安全稽核人員執行，但若有員工具備安全專業知識並熟悉內部標準，則這些員工也可以執行稽核的作業。

您可根據任何已知的商業風險指標，對專案進行優先順序排列，並對稽核序列進行分類，讓高風險軟體可受到較優先而頻繁的稽核。另外，低風險的專案可採用較寬鬆的內部稽核需求，以讓稽核實作更節省成本。

整體來說，每一個進行中的專案至少都應該每半年接受一次稽核。一般而言，若初始稽核後對應用程式保留了足夠的稽核資訊，後續稽核就會比較簡單。

對業主與利害關係人通告此服務，讓他們也可以對其專案要求稽核。內部標準中每項需求的及格/不及格詳細結果，都應該傳送給專案利害關係人進行評估。如果可行，稽核結果也應該含有衝擊解釋與補救建議。

標準與法規遵從

SC 3

對組織整體的標準與政策進行法規遵從需求制訂與專案衡量。

活動

建立專案的法規遵從標準

一旦組織建立了安全內部標準，要執行的下一個層級就是在產品生命週期中設定特定的標準點，專案唯有經內部標準稽核發現完全遵從法規，才可以及格通過。

法規遵從門檻的設定通常是在軟體版本發行時，只有在軟體通過了法規遵從檢查後才可以獲准發表。務必保留足夠的時間進行稽核以及矯正，稽核一般來說要早一點開始，例如在軟體某一版本交付 QA 時就可開始。

儘管有嚴密的法規遵從門檻，舊專案或其他特殊專案還是可能無法遵從，所以也必須建立例外核准的程序。全部專案中，可以獲得例外核准的專案不得超過 20%。

採用稽核資料收集的解決方案

組織若有定期進行專案團隊稽核，會隨著時間產生大量的稽核資料。此時應該使用自動化以協助自動化的收集、管理儲存與擷取的定序 (collation)，並限制個人對敏感稽核資料的存取。

對於來自內部標準的許多具體需求而言，您可自訂現有工具（例如：程式碼分析器、應用程式滲透測試工具、監控軟體等），用來根據內部標準進行自動化法規遵從檢查。自動化法規遵從檢查的目的在於改善稽核效率，並且讓更多的員工可以在進行正式稽核前，先執行法規遵從的自我檢查。另外，自動化檢查較不易出錯，也可提高發現問題的機率。

資訊儲存功能可讓您集中存取目前與過往的專案稽核資料。自動化解決方案也必須提供詳細的控制功能，限制唯有具備正當商業目的的核可人士，才能存取稽核資料。

所有與存取法規遵從資料以及要求存取權限的指示與程序，都應該通告給專案團隊。安全稽核人員一開始可能需要花多一點時間協助專案團隊。

結果

- 可從組織層級看見未遵循法規但卻被接受的風險
- 將專案層級的法規遵從保證性具象化
- 精確的追蹤以往的專案法規遵從歷史記錄
- 高效率的稽核程序可運用工具以減少人力支出

其他成功指標

- 超過 80%的專案已由稽核檢查確認遵從內部標準
- 與手動稽核比較，各項稽核時間減少 50%以上

其他成本

- 執行內部標準稽核自動化的建置或授權工具
- 持續維護稽核門檻與例外程序

其他人事

- 開發人員(1 天/年)
- 架構人員(1 天/年)
- 管理人員(1 天/年)

相關目標

- 〈學習與指導方針〉-3
- 〈程式碼檢閱〉-2
- 〈安全測試〉-2

相關法規遵從

策略性規劃

〈策略性規劃〉的功能著重於在組織內建立骨架，以進行軟體安全保證性方案。這是定義安全目標的最基礎步驟，並且配合了組織真正的商業風險。

組織會從小量的風險概況開始，隨著時間慢慢地為其應用程式與資料資產發展出更進階的風險分類方案。組織可以配合相關風險衡量的其他洞察力，調整其專案層級的安全目標並開發精細方向藍圖，讓安全方案更具效率。

在此功能內的更進階層級，組織可以利用許多內部與外部的資料來源，收集安全方案的指標與質化回應。這樣可以微調方案層級的成本花費與實際利益問題。

	SP 1	SP 2	SP 3
目標	**為組織內軟體安全建立統一的策略性方向藍圖。**	**瞭解資料與軟體資產的相關價值，並選擇風險承受度。**	**配合相關商業指標與資產價值調整安全支出。**
活動	預估整體商業風險概況 建置並維護保證性方案方向藍圖	根據商業風險對資料與應用程式進行分類 建立每個分類的安全目標	定期進行全業界的成本比較 收集過往安全花費的指標
結果	軟體所造成的最嚴重商業層級風險之具體清單 根據貴組織的安全需求，為您量身訂做額外負荷最小的方向藍圖 對於保證性方案的長期成長方向，具備整體組織層面的瞭解	各專案已根據商業的核心價值，自訂其保證性方案 對於資料與應用程式資產的安全相關事項，具備整體組織層面的瞭解 利害關係人可得到較完整的資訊以瞭解與接受風險	各專案安全支出決策需公告的相關事項 過去由於安全問題所致損失的估計 對各專案所做的安全支出與可能損失等問題考量 安全相關事宜的全業界事前審查

策略性規劃

SP 1

為組織內軟體安全建立統一的策略性方向藍圖。

活動

預估整體商業風險概況

與業主以及利害關係人進行會談，並將組織內的各種應用程式與資料資產所遇到的最惡劣情境列成清單。這份最惡劣情境清單會根據貴組織建立、使用或銷售軟體的方式而具有極大的變化、但是常見的問題包括：資料竊取或損毀、服務中斷、財務損失、反向工程、帳戶洩漏等等。

在廣泛地收集可能發生的最惡劣情境後，請根據收集到的資訊與核心商業價值相關知識，對此清單進行校對並選取最為嚴重的情境。您可以選取任意數量的情境，但是請盡量保持在 3 至 7 個選項，如此才可有效利用時間並維持練習的重點。

請詳細說明每一個選取的項目，並且仔細記錄造成最惡劣情境的因素、潛在的肇因以及對組織的潛在減緩因素。

最終的商業風險概況應該由業主與其他利害關係人檢閱，以更進一步的瞭解情境。

建置並維護保證性方案方向藍圖

瞭解組織所會遇到的主要商業風險後，請評估組織對此章中所提到的 12 項「功能」的目前表現。如果貴組織通過了所有的累計成功指標，請根據相對應的「目標」，以 1、2 或 3 對每項「功能」評分。如果未達到成功指標，請對該「功能」給予 0 的評分。

一旦充分瞭解了目前狀態，下一個目標就是辨識在下一個進度中所要改善的「功能」。選取的標準在於商業風險概況、其他的商業驅動因子、法規遵從需求、預算承受度等等。一旦選取「功能」之後，下一個進度的目標就是達到該功能之下的「目標」。

保證性方案的改善時間約需 3 至 6 個月，但至少每 3 個月就該進行一次保證性策略階段作業，以檢閱活動進度、對成功指標的表現以及其他可能需要進行方案變更的商業驅動因子。

結果

- 軟體所造成的最嚴重商業層級風險之具體清單
- 根據貴組織的安全需求，為您量身訂做額外負荷最小的方向藍圖
- 對於保證性方案的長期成長方向，具備整體組織層面的瞭解

成功指標

- 過去 6 個月內，有超過 80%的利害關係人接受了商業風險概況簡報
- 過去 3 個月內，有超過 80%的員工接受過保證性方案方向藍圖的簡報
- 過去 3 個月中已進行超過 1 項保證性方案策略階段作業

成本

- 商業風險概況建置與維護
- 對保證性方案進行每季評估

人事

- 開發人員(1 天/年)
- 架構人員(4 天/年)
- 管理人員(4 天/年)
- 業主(4 天/年)
- QA 測試人員(1 天/年)
- 安全稽核人員(4 天/年)

相關目標

- 〈標準與法規遵從〉-1
- 〈威脅模型分析〉-1
- 〈安全需求〉-2

相關法規遵從

SP 2

策略性規劃

瞭解資料與軟體資產的相關價值，並選擇風險承受度。

結果

- 各專案已根據商業的核心價值，自訂其保證性方案
- 對於資料與應用程式資產的安全相關事項，具備整體組織層面的瞭解
- 利害關係人可得到較完整的資訊以瞭解與接受風險

其他成功指標

- 過去 6 個月內，有超過 90%的應用程式與資料資產已進行了風險分類的評估
- 過去 6 個月內，有超過 80%的員工已接受了相關應用程式與資料風險分級的簡報
- 過去 3 個月內，有超過 80%的員工已接受過相關保證性方案方向藍圖的簡報

其他成本

- 應用程式與資料風險分類方案的建置或授權
- 更精細方向藍圖計畫所帶來的方案負荷

其他人事

- 架構人員(2 天/年)
- 管理人員(2 天/年)
- 業主(2 天/年)
- 安全稽核人員(2 天/年)

相關目標

- 〈標準與法規遵從〉-2
- 〈威脅模型分析〉-2
- 〈架構檢閱〉-2

相關法規遵從

活動

根據商業風險對資料與應用程式進行分類

建立簡易的分類系統以代表應用程式的風險層級。分類方式可以用「高/中/低」這種最簡易的格式來表示。您也可以使用比較精細的分類方式，但最好不要超過 7 個類別，且這些類別應大略地由高至低的漸層評分表示對商業風險的衝擊程度。

由組織的商業風險概況開始作業，建立可為各專案評定風險類別的專案評估標準。請另外為資料資產建立一個類似但獨立的分類方案，而每個項目都應該進行加權並根據對商業風險的潛在衝擊進行分類。

評估已收集的每項應用程式的資訊，並根據整體評估標準與使用中的資料資產風險分類，為每項資訊分派風險分類。這項工作可以由安全團隊集中進行，或由個別專案團隊透過自訂的問卷收集必要資訊後再進行。

您應該持續進行應用程式與資料資產風險分類的工作，為新的資產分配類別，並至少每半年更新一次現有資訊。

建立每個分類的安全目標

組織應用程式組合建置了分類方案後，就可以做出更精細的直接安全目標與保證性方案方向藍圖選擇。

您應修改保證性方案方向藍圖，指定各類別中特定「功能」的重點，使其能夠說明各應用程式的風險類別。對於保證性方案各項進度而言，這樣做通常會將比較高層級的「目標」優先排序在最高風險的應用程式層級，並且漸序地將比較不迫切的「目標」置於較低/其他的分類中。

這個程序會建立起組織的風險承受度，使您能夠在必須判斷各風險類別中應用程式的特定預期「目標」時，做出有效的決定。選擇讓較低風險應用程式在「安全功能」方面維持較低效能等級，就可省下資源以交換對加權風險的接受性。但您不需要刻意為每一項風險分類建立獨立的方向藍圖，因為這會導致保證性方案本身的管理缺乏效率。

策略性規劃

SP 3

配合相關商業指標與資產價值調整安全支出。

活動

定期進行全業界的成本比較

從業界內的溝通論壇、商業分析師以及顧問公司或是其他外部資源，對安全成本進行研究與資訊收集。並特別注意需辨識出來的少數關鍵因素。

首先，利用已收集的資訊，依照業界內相似類型的組織，辨識在安全支出所花費的平均金額。若要完成上述工作，您可以透過「由上而下」(Top-Down) 的方式，評估預算、收入等項目的總百分比；或者透過「由下而上」(Bottom-Up) 的方式，辨識在貴組織類型中屬於一般安全相關活動。整體而言，某些業界很難去進行這項衡量的工作，所以請儘量從可得的眾多相關資源中收集資訊。

研究安全成本的下一個目標，是要判斷貴組織目前使用的第三方安全產品與服務中，是否有節省成本的可能性。當衡量更換供應商的決定時，亦請考量重新訓練員工或其他方案負載等隱藏成本。

整體而言，開始進行後續保證性方案策略階段作業之前，請至少每一年進行一次此成本比較練習。比較資訊則應該上呈給利害關係人，讓他們根據業務對保證性方案進行更妥善的調整。

收集過往安全花費的指標

收集與過往安全意外成本有關的專案特有資訊。舉例來說，清除安全缺口所花費的時間與金錢、系統中斷所造成的財務損失、繳予法規機關的罰金與費用、專案特有工具或服務的一次性安全支出等等。

使用應用程式風險類別與其各自的指定保證性方案方向藍圖，就可以從與相對應風險分類相關的成本中，獲得每項應用程式基線安全成本的初始預估。

結合應用程式特有成本資訊，以及以風險類別為基礎的一般成本模型，再評估專案的離群值，例如；與風險分級不成比例的總額。這些值代表風險評估/分類出錯，或是表示必須調整組織的保證性方案，才能更有效率的解決安全成本的根源肇因。

保證性方案策略階段作業過程中，每季必須追蹤各專案的安全花費，而利害關係人每年應檢閱並評估所追蹤的資訊。對於離群值與其他無法預見的成本，應探討其對保證性方案方向藍圖的潛在影響。

結果

- 各專案安全支出決策需公告的相關事項
- 過去由於安全問題所致損失的估計
- 對各專案所做的安全支出與可能損失等問題考量
- 安全相關事宜的全業界事前審查

其他成功指標

- 過去 3 個月內，有超過 80%的專案曾報告過安全成本
- 過去 1 年中曾進行 1 次以上的全業界成本比較
- 過去 1 年中曾進行 1 次以上的過往安全花費評量

其他成本

- 安全方案的業界智慧(industry intelligence) 建置或授權
- 來自成本預估、追蹤以及評估的方案負荷

其他人事

- 架構人員(1 天/年)
- 管理人員(1 天/年)
- 業主(1 天/年)
- 安全稽核人員(1 天/年)

相關目標

- 〈弱點管理〉-1

相關法規遵從

威脅模型分析

〈威脅模型分析〉的功能，主要著重在根據開發中軟體的商業功能辨識與瞭解專案層級風險。 組織可從威脅相關細節與對每項專案的可能攻擊中，透過對安全開發案的優先順序，做出更理想的決定，以讓組織整體運作更有效率。另外，對於風險承受度的決定是在獲得良好資訊的情況下做出的，所以也比較符合商業實際情形。

組織可藉由一開始的簡易攻擊樹狀結構(attack tree)，建立更詳細的威脅檢查與加權方式，隨著時間過去而獲得改善。最後，一個分工嚴謹的組織會緊密結合補償因素與來自外部實體的呈報風險，進行此資訊的維護。 這樣可更廣泛的瞭解安全問題的潛在下游衝擊，同時也可密切觀察組織對已知威脅的目前表現。

	TM 1	TM 2	TM 3
目標	**識別並瞭解對組織與個別專案的高層級威脅。**	**提高威脅評估的準確度，並改善對每一件專案瞭解的細膩程度。**	**對內部與第三方軟體中的各個威脅，具體的做好補償控制。**
活動	建立並維護每個應用程式的攻擊樹狀結構 從軟體架構開發攻擊者概況	配合應用程式特有資料詳細建立攻擊樹狀結構 採用威脅測量的加權系統	確切地評估第三方元件的風險 配合補償控制項詳細建立攻擊樹狀結構
結果	對可能導致負面結果的因素擁有高層級的瞭解 加強對專案團隊之間的威脅警覺性 清查貴組織受到的威脅	對個別專案可能發生的威脅有精細的瞭解 能讓專案團隊建立更佳得失抵換決策的骨架 根據風險加權優先排序專案團隊內的開發工作	更進一步考量每項軟體專案的完整威脅概況 仔細的將保證功能配對至每項軟體專案的已確立威脅 根據每個軟體專案的商業功能，進行文件的事前審查

威脅模型分析

TM 1

識別並瞭解對組織與個別專案的高層級威脅。

活動

建立並維護每個應用程式的攻擊樹狀結構

僅根據每個軟體專案的商業目標以及商業風險概況（如有的話），辨識出每個專案團隊開發中軟體的最惡劣情境。用一句話就識別出各種最惡劣情境，並將其標示為攻擊者的高層級目標。

從每項已辨識的攻擊者目標中，辨識出要實現每個目標所必要的保留指定條件。此資訊應該擷取自每項目標下的分支中，而每個分支則為其下所包含的邏輯 AND 或邏輯 OR 陳述式。AND 分支表示每個直接連接的子節點都必須為 True，以實現父節點。OR 分支表示任一個直接連接的子節點都必須為 True，以達到父節點。

請檢閱每項目前以及過往的各功能需求（如有的話），並且擴大攻擊樹狀結構以指出每項需求相關的安全失敗。透過腦力激盪的方式，將每個節點不斷細分成分支，指出攻擊者達到其中一項目標的所有可能途徑。初始建立應用程式的攻擊樹狀結構之後，當軟體進行重大變更時就必須連帶更新此樹狀結構，以消除節點或新增節點。這項評估的工作必須由資深開發人員或架構人員進行，或者是由一或多位安全稽核人員進行。

從軟體架構開發攻擊者概況

請在一開始就進行評估，根據軟體專案辨識出可能對組織造成影響的所有威脅。針對這項評估，將威脅的範圍限制在具有惡意意圖的代理程式，並忽略其他如已知弱點、潛在弱點等風險。

評估一開始可以大致考量外部代理程式，以及其對應的攻擊動機。在此清單上新增會造成損害的內部角色，以及進行內部攻擊的動機。根據考量中軟體專案的架構，比較有效率的作法是對每種架構類型進行一次這種分析即可，而不需要針對個別專案進行分析，因為應用程式的架構與商業目的大致上會受到類似威脅。

此分析應該配合業主與其他利害關係人一起進行，但也應該包括一位或多位安全稽核人員以對威脅進行其他的檢視。最終的目標是完成一份簡易清單，列出威脅代理程式與對應的攻擊動機。

結果

- 對可能導致負面結果的因素擁有高層級的瞭解
- 加強對專案團隊之間的威脅警覺性
- 清查貴組織受到的威脅

成功指標

- 過去 12 個月內，有超過 50%的專案利害關係人接受過相關專案的攻擊樹狀結構簡報
- 超過 75%的專案利害關係人接受過相關架構的攻擊者概況簡報

成本

- 攻擊樹狀結構專案成品的建置與維護

人事

- 業主(1 天/年)
- 開發人員(1 天/年)
- 架構人員(1 天/年)
- 安全稽核人員(2 天/年)
- 管理人員(1 天/年)

相關目標

- 〈策略規劃〉-1
- 〈安全需求〉-2

相關法規遵從

TM 2

威脅模型分析

提高威脅評估的準確度，並改善對每一件專案瞭解的細膩程度。

結果

- 對個別專案可能發生的威脅有精細的瞭解
- 能讓專案團隊建立更佳得失抵換決策的骨架
- 根據風險加權優先排序專案團隊內的開發工作

其他成功指標

- 超過 75%的專案團隊已辨識與評等威脅
- 過去 6 個月內，有超過 75%的專案利害關係人接受過相關專案的攻擊樹狀結構簡報
- 過去 12 個月中，專案攻擊樹狀結構已辨識出超過 50%的安全事件

其他成本

- 維護攻擊樹狀結構與攻擊者概況所造成的專案負荷

其他人事

- 安全稽核人員(1 天/年)
- 業主(1 天/年)
- 管理人員(1 天/年)

相關目標

- 〈策略規劃〉-2
- 〈防禦設計〉-2

相關法規遵從

-

活動

配合應用程式特有資料詳細建立攻擊樹狀結構

詳細分析每個軟體專案的已建立攻擊樹狀結構，以涵蓋軟體設計與實施有關的技術細節。若要完成這項工作，需考量軟體的相關眾多因素。大型應用程式的攻擊樹狀結構相當冗長，如有需要請進行分割，但也請確保分割的每一部份都含有根目標節點的複本，以及路徑上的連接節點。

一開始請使用在設計階段中所考量的安全相關假設進行，並使用說明每項假設失敗的節點，擴大攻擊樹狀結構。根據專案的安全需求，詳細說明攻擊樹狀結構上的節點，指出每個安全需求失敗的必要先決條件。另外，說明與軟體專案有關的任何已知安全事件，以確保事件已顯示在攻擊樹狀結構上。

最後，辨識使用者角色以及其對應能力，以更進一步詳細解釋在攻擊樹狀結構中所說明的內部攻擊情境。如果可行的話，請使用應用程式特有的權限矩陣，構想責任劃分考量等失敗情境。

採用威脅測量的加權系統

根據已建立的攻擊者概況，辨識評比系統以允許在威脅之間進行相關的比較。一開始，這只會是一種根據商業風險所做的「高-中-低」簡易評比，但只要量表不超過五個分類，您就可以使用該量表。

在辨識評比系統之後，請建立評估標準以評比每個威脅。為了正確進行評比，必須考量每個威脅動機以外的因素。重要的因素包括資金與人力資源、原有存取權限、技術能力、樹狀結構上的相關目標、攻擊成功的可能性以及其他因素。

在進行每種威脅的評比之後，請使用此資訊將開發生命週期內的風險減緩活動進行優先排序。一旦為某一專案團隊建立起此系統後，應該在設計新功能或是執行重整時更新此評比系統。

威脅模型分析

TM 3

對內部與第三方軟體中的各個威脅，具體的做好補償控制。

活動

確切地評估第三方元件的風險

評估您的軟體程式碼知識庫，並且辨識所有外來元件。一般而言，這些元件包括開放來源專案、購買的 COTS 軟體以及軟體所使用的線上服務。

針對每個已辨識的元件，根據第三方元件潛在的洩露問題，詳細說明軟體專案的攻擊者概況。請根據新辨識的攻擊者概況，更新攻擊樹狀結構，以整合所有根據新攻擊者目標或能力所產生的可能風險。

除了威脅情境外，也請考量第三方軟體的弱點或設計缺陷，會以哪些方式影響您的程式碼與設計。請配合已更新攻擊者概況的弱點與知識，確實的詳細製作攻擊樹狀結構。

首次評估專案後，必須在設計階段或每一個開發週期中，更新與檢閱此樹狀結構。這項活動應該由安全稽核人員，配合相關技術人員與商業利害關係人一起進行。

配合補償控制項詳細建立攻擊樹狀結構

進行評估以正式辨識可直接防止洩露先決條件的因素，在攻擊樹狀結構上所代表的節點。這些減緩因素為補償控制項，可正式解決來自軟體的風險。因素可為軟體本身的技術功能，但也可以是開發生命週期、基礎架構功能等項目內的程序元素。

各分支會根據攻擊樹狀結構上的邏輯關係，代表 AND 或 OR 陳述式。因此，若減緩 AND 分支上的某先決條件，則父節點與連接的全部分葉節點都可標示為已減緩。但是，對於 OR 節點，則需先避免節點上的全部子節點，才能將父節點標示為已減緩。

您可以藉由檢閱攻擊樹狀結構，找出可將最多節點標示為已減緩的補償控制項。請針對從分頁節點到頂端層級目標的任何可用路徑，辨識對組織策略回應的潛在補償控制項。

首次評估專案後，必須在設計階段或每一個開發週期中，更新與檢閱此樹狀結構。這項活動應該由安全稽核人員，配合相關技術人員與商業利害關係人一起進行。

結果

- 更進一步考量每項軟體專案的完整威脅概況
- 仔細的將保證功能配對至每項軟體專案的已確立威脅
- 根據每個軟體專案的商業功能，進行文件的事前審查

其他成功指標

- 在每項實施週期之前，都有超過 80%的專案團隊已備妥更新的攻擊樹狀結構
- 在每次發行新版本之前，都有超過 80%的專案團隊已備妥更新的第三方元件詳細目錄

其他成本

- 維護詳細攻擊樹狀結構與擴充的攻擊者概況所造成的專案負荷
- 發現所有第三方的相依性

其他人事

- 業主(1 天/年)
- 開發人員(1 天/年)
- 架構人員(1 天/年)
- 安全稽核人員(2 天/年)
- 管理人員(1 天/年)

相關目標

- 〈安全需求〉-2 和 3

相關法規遵從

-

安全需求

〈安全需求〉的功能著重於主動指定軟體在安全方面的預期行為。透過專案層級的其他分析活動，就會根據軟體的高層級商業目標開始收集安全需求。隨著組織日益進步，就會使用更精密的技術（例如；不當使用案例模型分析）來發現開發初期難以察覺的新安全需求。

此功能佈建的形式非常精密，並可詳細說明如何將組織的安全需求導入與供應商之間的關係，再稽核專案以確保一切符合安全需求規格相關預期。

	SR 1	SR 2	SR 3
目標	**考量軟體需求程序中軟體的安全明確性。**	**根據應用程式特有風險與濫用情況產生安全需求。**	**規定所有軟體專案與第三方相依性的廣泛安全需求程序。**
活動	從商業功能中衍生安全需求 使用指導方針、標準與法規遵從資源	建立與維護每個專案的不當使用案例模型 根據已知風險指定安全需求	將安全需求加入供應商協議內容中 擴展安全需求的稽核方案
結果	開發成果高度配合商業風險 特地取得業界最佳安全實作，以做為確切的需求 利害關係人能瞭解減緩軟體風險的程度	對商業邏輯的攻擊情境有著詳細的理解 根據可能攻擊，優先排序安全功能的開發工作 對功能與安全人力的得失抵換之間，可做出更有經驗的決策 利害關係人可以更有效的避免原本具有安全缺陷的功能需求	正式為外來程式碼的安全預期設定基線 獲得每個專案團隊進行中安全工作的集中資訊 根據應用程式風險與希望的安全需求，具備調整資源供專案使用的能力

安全需求

SR 1

考量軟體需求程序中軟體的安全明確性。

活動

從商業功能中衍生安全需求

檢閱指定商業邏輯與每個軟體專案整體行為的功能需求。收集專案的需求後，進行評估以衍生相關的安全需求。即使軟體由第三方所建立，需求一經辨識後，就應該包括在功能需求中以交付供應商。

安全稽核人員應該針對每個功能需求，指引利害關係人瞭解全部程序，並明確地指出與安全相關的預期。一般而言，澄清每個需求的問題包括了對資料安全、存取控制、交易完整性、商業功能的關鍵性、責任劃分以及其他項目的預期。

請務必確定全部的安全需求均遵循相同的原則，以便一致撰寫出完善的需求。需求必須力求明確、可衡量以及合理。

進行中專案的所有新需求都必須依此程序進行。針對現有功能，我們建議進行相同的程序做為間隔分析，以利未來安全重構工作。

使用指導方針、標準與法規遵從資源

判斷專案團隊應納入需求的業界最佳實作。您可從公開使用的指導方針、內部或外部標準或者已建立的法規遵從需求中，選擇上述最佳實作。

請勿在各項開發階段中包含太多最佳實作需求，因為設計與實施兩者所需的時間會互相排擠。我們建議的方式是，慢慢地在連續開發週期中加入最佳實作，隨著時間演進逐漸加強軟體的整體保證概況。

針對現有系統重構安全最佳實作是一項複雜的工作。您可適時在新增功能時增加安全需求。請至少進行一次分析，辨識可應用的最佳實作以供未來規劃工作使用。

這項檢閱的工作應該由安全稽核人員進行，並且獲得商業利益關係人的參與。資深開發人員、架構人員以及其他技術利害關係人都應參與這項工作，將設計與實施特有的知識帶入決策程序。

結果

- 開發成果高度配合商業風險
- 特地取得業界最佳安全實作，以做為確切的需求
- 利害關係人能瞭解減緩軟體風險的程度

成功指標

- 超過 50%的專案團隊擁有明確定義的安全需求

成本

- 將新增安全需求加入每個開發週期所造成的專案負荷

人事

- 安全稽核人員(2 天/年)
- 業主(1 天/年)
- 管理人員(1 天/年)
- 架構人員(1 天/年)

相關目標

- 〈學習與指導方針〉-1
- 〈標準與法規遵從〉-2
- 〈架構檢閱〉-1
- 〈程式碼檢閱〉-1
- 〈安全測試〉-1

相關法規遵從

SR 2

安全需求

根據應用程式特有風險與濫用情況產生安全需求。

結果

- 對商業邏輯的攻擊情境有著詳細的理解
- 根據可能攻擊，優先排序安全功能的開發工作
- 對功能與安全人力的得失抵換之間，可做出更有經驗的決策
- 利害關係人可以更有效的避免原本具有安全缺陷的功能需求

其他成功指標

- 過去 6 個月內，有超過 75%的專案已更新不當使用案例模型

其他成本

- 不當使用案例模型的建置與維護所造成的專案負荷

其他人事

- 安全稽核人員(2 天/年)
- 管理人員(1 天/年)
- 架構人員(2 天/年)
- 業主(1 天/年)

相關目標

- 〈威脅模型分析〉-1 和 3
- 〈策略規劃〉-1

相關法規遵從

-

活動

建立與維護每個專案的不當使用案例模型

更進一步考量軟體專案的功能需求，進行更正式的分析以判斷潛在誤用或不當使用功能。一般而言，此程序會始於辨識正常使用情境，例如；使用案例圖（如有的話）。

您可以從一般使用情況的陳述開始，然後以腦力激盪的方式思考該陳述遭到全部或部分否定的情況，以針對各種使用情境產生不當使用案例集。最簡易的開始方式就是在使用陳述式中以各種方式插入「no」或「not」兩字，通常是在名詞或動詞周圍插入。每種使用情境應該可產生數個可能的不當使用案例陳述式。

請更進一步詳細製作不當使用案例的陳述式，根據軟體的商業功能，包含各種應用程式特有的考量。最終目標是獲得完整的不當使用陳述式集，以形成軟體所不允許的使用模式模型。

首次建立不當使用案例模型後，在設計階段中應針對進行中專案更新此模型。請針對現有專案分析新需求，以判斷是否有潛在的不當使用，而現有的專案應該針對實務的已建立功能，適時建立不當使用案例。

根據已知風險指定安全需求

明確的檢閱現有成品，指出組織或專案的特有安全風險，以更加瞭解軟體的整體風險概況。若可能的話，也可利用如高層級商業風險概況、個別應用程式攻擊樹狀結構等資源，以及來自架構、程式碼檢閱、安全測試等的數據。

除了檢閱現有成品外，請使用應用程式的不當使用案例模型，以辨識直接或間接減緩不當使用情境的具體安全需求。

此程序應依需要由業主與安全稽核人員。最後，風險帶來新安全需求的概念，應該成為計劃階段中的必要步驟，新發現的所有風險都要由專案團隊特定進行評估。

安全需求

SR 3

規定所有軟體專案與相依性的廣泛安全需求程序。

活動

將安全需求加入供應商協議內容中

除了由先前分析辨識出的安全需求類型之外，可從第三方協議中衍生其他安全效益。一般而言，組織內部會進行需求 (或者是高層級設計) 的開發，而低層級設計與實施通常則由供應商負責。

請根據每項外部開發元件的特定分工，辨識特定的安全活動與技術評估標準，以便加入與供應商的合約中。普遍而言，這應該是一項來自「驗證與評估專業領域」的活動集，內容涵蓋〈架構檢閱〉、〈程式碼檢閱〉以及〈安全測試〉功能。

外部開發的元件使用方式

因為新增其他的需求都會提高成本，所以與每個供應商的協議語言應該根據個別案例進行修改。每項潛在安全活動的成本應考量每個元件或系統的使用量，對活動的效益進行收支平衡的計算。

擴展安全需求的稽核方案

將安全需求完整度的檢查整合至定期專案稽核中。因為沒有專案特有知識就很難進行測量，所以稽核應該專注在檢查如需求或設計文件等專案成品上，以證明進行了正確的分析類型。

特別是每個功能需求都應該根據商業驅動因子與預期的不當使用情境，附註安全需求。整體專案需求應該包含由指導方針與標準產生的最佳實作需求清單。另外也應該有一份未完成安全需求的清單，以及要在未來版本發行佈建的預估時間表。

此稽核應該在每項開發階段中執行，理想的執行時間是在需求程序接近尾聲時，但一定要在軟體發行前進行稽核。

結果

- 正式為外來程式碼的安全預期設定基線
- 獲得每個專案團隊進行中安全工作的集中資訊
- 根據應用程式風險與希望的安全需求，具備調整資源供專案使用的能力

其他成功指標

- 過去 6 個月內，有超過 80%的專案通過安全需求稽核
- 過去 12 個月內，有超過 80%的供應商協議進行過合約的安全需求分析

其他成本

- 其他安全需求委外開發造成的成本增加
- 安全需求發行門檻所產生的進行中專案管理費

其他人事

- 安全稽核人員(2 天/年)
- 管理人員(2 天/年)
- 業主(1 天/年)

相關目標

- 〈威脅模型分析〉-3
- 〈標準與法規遵從〉-2

相關法規遵從

防禦設計

〈防禦設計〉功能著重於說明組織要建立預設安全軟體的主動步驟。建立出無下游弱點的軟體，就可大幅降低軟體的整體安全風險。

組織可根據最佳實作，從高層級設計指導方針開始，朝著一貫使用的設計模式演進以達成安全功能。然後這些活動也能夠以此概念為中心，讓專案團隊更加瞭解商業邏輯，並建立較正式的方法以確保軟體首次設計的安全。

隨著組織日益改進，此功能的佈建方式會日趨精密，也讓組織建立起參考平台，說明組織所建立的一般軟體類型。這個參考平台可當作骨架，讓開發人員以較少的弱點風險建立自訂軟體。

目標	**為專案團隊提供主動設計指導方針，並加強周邊安全。**	**在設計程序內，建立具備完整安全加強功能的軟體。**	**衡量專案團隊的主動能力，讓防禦設計更有效率。**
活動	維護建議軟體骨架的清單 明確地將安全原則套用至周邊介面	建立資源存取的權限矩陣 從架構辨識安全設計模式	建立正式參考平台 驗證骨架、模式以及平台的使用
結果	未預期相依性之特定阻礙與一次性實施選擇 利害關係人瞭解因選定的資源庫與骨架而提高的專案風險 建立了開發過程中的協定，可主動將安全機制套用至設計	詳盡的規劃資產與使用者角色的配對，以在設計中做出更好的區隔 可重複使用的設計建置區塊，以佈建安全保護與功能 使用已建立的安全設計技術，增加軟體專案的可信度	自訂應用程式開發平台，提供內建安全保護 全組織預期開發過程中的主動安全工作 利害關係人可根據安全設計的商業需要，建立更佳的得失抵換決策
	DD 1	DD 2	DD 3

防禦設計

DD 1

為專案團隊提供主動設計指導方針，並加強周邊安全。

活動

維護建議軟體骨架的清單

辨識出組織內所有軟體普遍使用的第三方軟體資源庫，以及使用中骨架。一般而言，您不需要徹底搜尋相依性，而請專注於擷取最常使用的高層級元件。

您可以根據第三方元件所提供的核心功能，將清單上的元件分類為不同的功能類別。您也需要備註每個元件在所有專案團隊中的使用普及率，以加權對第三方程式碼的依賴性。請將此加權清單作為指南，建立一份元件清單並通告組織內開發人員將其視為建議元件。

建議清單上的元件可由數個因素來決定。雖然您可以不需透過特定的搜尋，就可建立這份清單，但我們仍建議您檢查每項元件的事件歷程記錄、對弱點的回應追蹤記錄、對組織的功能適用性，以及第三方元件使用上的過度複雜性等項目。

這份清單應由資深開發人員與架構人員建立，但是也需要管理人員與安全稽核人員的參與。建議元件與功能分類配對的清單建立之後，應該將此清單通告給組織內的開發人員。最後的目標是提供已詳細分析過的預設項目給專案團隊。

明確地將安全原則套用至周邊介面

在設計的過程中，專案團隊的技術人員應使用安全原則的簡短指導清單，做為系統邊緣介面的檢查清單。一般而言，安全原則包括縱深防禦 (defense in depth)、保護最弱的連結 (securing the weakest link)、使用安全預設項 (use of secure defaults)、安全功能的簡潔設計(simplicity in design of security functionality)、安全故障 (secure failure)、安全與可用性之間的平衡 (balance of security and usability)、以最低權限執行 (running with least privilege)，以及避免透過隱匿來實現安全 (avoidance of security by obscurity)等。

設計團隊應該針對每個已知的周邊介面，以整體系統為前提來考量每項原則，並辨識可加入的功能以提升該介面的安全。一般而言，應該將這些功能限制為只佔用其他工作的一小部份，而不會超過功能需求的正常實施成本，並且要記下所有較大型的功能並排程於未來版本發行。

這項程序應該由每個接受過安全警覺訓練的專案團隊進行，但如果有更多的專業安全員工在旁協助，將非常有助於設計決策的制定。

結果

- 未預期相依性之特定阻礙與一次性實施選擇
- 利害關係人瞭解因選定的資源庫與骨架而提高的專案風險
- 建立了開發過程中的協定，可主動將安全機制套用至設計

成功指標

- 過去 1 年中，有超過 80%的開發人員接受了軟體骨架建議的簡報
- 超過 50%的專案已自我報告需設計的安全原則應用程式

成本

- 軟體骨架建議的建置、維護以及警覺
- 安全原則分析與應用程式所所產生的進行中專案管理費

人事

- 架構人員(2 至 4 天/年)
- 開發人員(2 至 4 天/年)
- 安全稽核人員(2 至 4 天/年)
- 管理人員(2 天/年)

相關目標

- 〈學習與指導方針〉-1

相關法規遵從

-

DD 2

防禦設計

在設計程序內，建立具備完整安全加強功能的軟體。

結果

- 詳盡的規劃資產與使用者角色的配對，以在設計中做出更好的區隔
- 可重複使用的設計建置區塊，以佈建安全保護與功能
- 使用已建立的安全設計技術，增加軟體專案的可信度

其他成功指標

- 過去 6 個月內，有超過 80%的專案已更新權限矩陣
- 過去 6 個月內，有超過 80%的專案團隊接受過可應用安全模式的簡報

其他成本

- 可應用安全模式的建置或授權
- 維護權限矩陣所產生的進行中專案管理費

其他人事

- 架構人員(2 至 4 天/年)
- 開發人員(1 至 2 天/年)
- 管理人員(1 至 2 天/年)
- 業主(1 天/年)
- 安全稽核人員(1 至 2 天/年)

相關目標

- 〈學習與指導方針〉-2

相關法規遵從

-

活動

建立資源存取的權限矩陣

根據應用程式的商業目的，辨識使用者與操作者角色。另外，藉由收集全部的相關資料資產，以及受某種存取控制形式保護的應用程式特有功能，以建立資源清單。

請在一個以角色與資源為兩軸的簡易矩陣內，考量每個角色與資源間的關係，並根據利害關係人的存取控制，在每個交集處備註系統的正確行為。

對於資料資源而言，請務必依據建立、讀取、更新以及刪除備註存取權。存取權層次對功能資源而言，很可能為應用程式特有功能，但至少要備註該角色是否獲准存取該功能。

此權限矩陣可當作一項成品，以記錄整體系統商業邏輯的正確存取控制權。權限矩陣應該由專案團隊與商業利害關係人一起建立。首次建立矩陣後，商業利害關係人應該在每次版本發行前更新矩陣，通常的更新時間是在設計階段開始之前。

從架構辨識安全設計模式

組織的每項軟體專案都應該依據一般架構類型進行分類。常見的分類包括主從式應用程式、內嵌系統、桌面應用程式、網頁介面應用程式、網路服務平台、交易中介軟體系統，以及大型主機應用程式等分類。您可以根據貴組織的專長，依程式語言、處理器架構或甚至部署時期來開發更細的分類。

對於一般軟體架構類型而言，一般設計模式集代表可衍生出完善的安全功能實施方法，並可套用至組織內軟體專案的個別設計上。這些安全設計模式代表了一般設計元素的普遍定義，而且可供研究或購買。如果這些模式已經自訂成符合貴組織的特定模式，將更有助於貴組織。範例模式包括單一簽入子系統、跨階層委派模型、強化介面設計、責任劃分授權模型，以及集中記錄模式等模式。

辨識可應用與適用模式的程序，應該由架構人員、資深開發人員以及設計階段中的其他技術利害關係人執行。

防禦設計

DD 3

衡量專案團隊的主動能力，讓防禦設計更有效率。

活動

建立正式參考平台

配合每種架構類型的特定安全模式作業之後，專案團隊就應該選出程式碼集合以實施各項功能，並做為共享程式知識庫的基礎。此共享程式知識庫可在初期做為每個專案必須使用的常用建議資源庫集合，然後會隨著時間漸漸成長為一項或多項軟體骨架，以代表專案團隊建立軟體的參考平台。參考平台的範例包括「模型-檢視-控制器」(MVC) 網路應用程式骨架、支援交易後端系統資源庫、網路服務平台骨架、主從式應用程式的鷹架，以及使用可插入商業邏輯的中介軟體架構等。

另一個建立初始參考平台的方法是，選取一個位於生命週期早期的特定專案，並要求專業安全員工一起作業，用一般的方式建立安全功能，如此就可從專案中擷取出來並利用於組織某處。

姑且不論建立的方式，參考平台的優勢在於加快稽核與安全相關檢閱的速度、增加開發效率以及降低維護其他負荷。

參考平台的設計與建立必須有架構人員、資深開發人員以及其他技術利害關係人的參與。建立平台後，必須有一組團隊持續維護支援與更新。

驗證框架、模式以及平台的使用

在定期專案稽核過程中，請執行專案成品其他分析，以衡量建議骨架、設計模式以及參考平台的使用。雖然是在定期稽核過程中進行此活動，然此活動的目標是從專案團隊盡量收集回應，以衡量個別團隊的主動安全工作。

整體而言，請務必與專案團隊驗證多項因素。辨識出使用非建議骨架的情形，以判斷建議與組織功能需求之間是否有差異。檢驗未用或誤用的設計模式與參考平台模組，判斷是否需要更新。另外，隨著組織日益精進，專案團隊可能會樂見參考平台可實施更多或不同的功能。

任何專業安全員工都可執行此分析。收集自各專案的指標，應該由管理人員與利害關係人進行校對以便分析。

結果

- 自訂應用程式開發平台，提供內建安全保護
- 全組織預期開發過程中的主動安全工作
- 利害關係人可根據安全設計的商業需要，建立更佳的得失抵換決策

其他成功指標

- 超過 50%的進行中專案使用了參考平台
- 過去 6 個月內，有超過 80%的專案報告了骨架、模式以及平台使用回應
- 超過 3.0 李克特量表得分，肯定專案團隊所報告的指導方針/平台效用

其他成本

- 建置或授權參考平台
- 持續維護與支援參考平台
- 稽核過程中，使用驗證所產生的進行中專案管理費

其他人事

- 管理人員(1 天/年)
- 業主(1 天/年)
- 架構人員(3 至 4 天/年)
- 開發人員(2 至 3 天/年)
- 安全稽核人員(2 天/年)

相關目標

- 〈標準與法規遵從〉-2
- 〈架構檢閱〉-3
- 〈程式碼檢閱〉-3
- 〈安全測試〉-3

相關法規遵從

架構檢閱

〈架構檢閱〉功能著重於檢視軟體設計以及架構模型是否有安全問題。這樣可允許組織及早在軟體開發過程中偵測架構層級問題，並因此避免重整所有程序所造成的潛在大幅成本。

一開始，您可進行少量的活動以瞭解架構的安全相關細節，再漸漸採取更正式的檢查方式，讓組織得以驗證安全機制佈建的完整性。架構檢閱服務建立於組織層級，可提供給利害關係人使用。

架構檢閱功能精密的格式包括詳細、資料層級的設計檢查，並在發行獲准前執行標準預期以獲得評估數據。

	AR 1	AR 2	AR 3
目標	**支援軟體架構的特定檢閱，以確保已知風險的基線減緩。**	**提供評估服務，以檢閱軟體設計是否符合完整的安全最佳實作。**	**要求評估與驗證成品，以更詳細地瞭解保護機制。**
活動	辨識軟體易受攻擊面 分析設計是否符合已知安全需求	檢查是否完整佈建安全機制 部署專案團隊的設計檢閱服務	開發敏感資源的資料流程圖 建立架構檢閱的版本發行門檻
結果	對周邊架構的安全含意有高層級的瞭解 讓開發團隊自我檢查，達到安全最佳實作的設計 以少量程序進行專案層級架構檢閱	正式提供評估服務以一致檢閱架構安全 精確找出維護模式與傳統系統中的安全缺陷 專案利害關係人更加瞭解軟體提供保證保護的方式	精密檢閱系統設計中的弱點，以利建立更優異的區間 達成組織層級警覺，判斷專案是否符合架構的基線安全預期 比較專案，有效率的減緩已知缺陷

架構檢閱

AR 1

支援軟體架構的特定檢閱，以確保已知風險的基線減緩。

活動

辨識軟體易受攻擊面

針對每項軟體專案，建立簡化的整體架構檢閱。一般而言，您應該根據專案成品來建立這項檢閱，而這些成品可為高層級需求與設計文件、與技術員工的會談，或是程式知識庫的模組層級檢閱。擷取系統內高層級模組的確非常重要，但就精細度上的優先準則而言，應先確定可在單一頁面上顯示檢閱中總體系統的資料圖表。

您可以從此單一頁面的架構檢閱，依據來自授權使用者、匿名使用者、操作者、應用程式特有角色等的介面存取性分析每項元件。您也應該以單一頁面檢閱來考量提供介面的元件，以在圖表上找出功能委派點或是對其他元件的資料呈報。將擁有類似存取性概況的介面與元件設為群組，並將此群組做為軟體易受攻擊面。

進一步為每個介面詳述此單一頁面的圖表，以備註所有的安全相關功能。根據組成易受攻擊面的已辨識介面群組，檢查模型的設計層級一致性，以瞭解如何保護具備類似存取的介面。一致性中的任何缺陷都可記載為已發現的評估問題。

無論是在專案團隊內或外部，都應該由專業安全技術員工執行這項分析。一般而言，初始建立圖表與易受攻擊面分析之後，只需要在新增或變更邊緣系統介面的設計階段時，更新此分析。

分析設計是否符合已知安全需求

不論是經過正式管道辨識獲得，或經由非正式管道得知的安全需求，都應該加以辨識與收集。另外，也請辨識並涵蓋系統安全作業所倚賴的所有安全假設。

對照系統架構的單一頁面圖表，檢閱已知安全需求清單上的每個項目。詳細製作圖表，顯示用來解決每項安全需求的設計層級功能。如果系統過於龐大與/或複雜，您可以另外製作一份精細的圖表，以簡化擷取此資訊的動作。這項分析的整體目標是，驗證系統設計已經解決了每項已知的安全需求。所有未在設計層級詳細說明的安全需求，都必須備註為評估所發現的問題。

這項分析應該配合架構人員、開發人員、管理人員以及必要業主的參與，由專業安全技術員工執行。安全需求或高層級系統設計進行變更時，您必須在設計階段中更新此分析。

結果

- 對周邊架構的安全含意有高層級的瞭解
- 讓開發團隊自我檢查，達到安全最佳實作的設計
- 以少量程序進行專案層級架構檢閱

成功指標

- 過去 12 個月內，有超過 50%的專案已更新易受攻擊面分析
- 過去 12 個月內，有超過 50%的專案已更新安全需求設計層級的分析

成本

- 建置與維護每項專案的架構圖表
- 檢查遭受攻擊面與安全需求設計所產生的進行中專案管理費

人事

- 架構人員(2 至 3 天/年)
- 開發人員(1 至 2 天/年)
- 管理人員(1 天/年)
- 安全稽核人員(1 天/年)

相關目標

- 〈安全需求〉-1

相關法規遵從

-

AR 2

架構檢閱

提供評估服務，以檢閱軟體設計是否符合完整的安全最佳實作。

結果

- 正式提供評估服務以一致檢閱架構安全
- 精確找出維護模式與傳統系統中的安全缺陷
- 專案利害關係人更加瞭解軟體提供保證保護的方式

其他成功指標

- 過去 6 個月內，有超過 80%的利害關係人接受過檢閱要求狀態的簡報
- 過去 12 個月內，有超過 75%的專案進行了架構檢閱

其他成本

- 建置、訓練以及維護設計檢閱團隊
- 檢閱活動所產生的進行中專案管理費

其他人事

- 架構人員(1 至 2 天/年)
- 開發人員(1 天/年)
- 管理人員(1 天/年)
- 安全稽核人員(2 至 3 天/年)

相關目標

- 〈學習與指導方針〉-2
- 〈策略規劃〉-2

相關法規遵從

-

活動

檢查是否完整佈建安全機制

針對高層級架構圖表內模組的每項介面，正式逐項檢查安全機制清單，並針對介面的佈建進行系統分析。這類型的分析應該在內部介面（例如：階層之間）以及外部介面（例如：組成易受攻擊面的介面）上執行。

您要考量的的六項主要安全機制為：驗證、授權、輸入驗證、輸出編碼、錯誤處理以及記錄等六項。也請考量相關的加密與階段作業管理機制。對每個介面，判斷系統設計上要放置機制的位置，並將所有缺少或有疑問的功能記載為發現問題。

這項分析應該由專業安全員工，在專案團隊的應用程式特有知識協助下執行。每發行一次軟體就應該執行一次這項分析，而執行時間通常可設在設計階段結束之前。初始建立此分析後，後續的版本發行必須根據開發週期中所做的變更，更新所發現的問題。

部署專案團隊的設計檢閱服務

設計一項程序，讓專案利害關係人可要求檢閱架構。這項服務可以由組織內部集中提供，或分配給現有員工執行，但全部的檢閱人員都必須接受訓練，以執行完整、一致的檢閱。

檢閱服務應予集中管理，而檢閱要求序列則應該由熟悉組織整體商業風險概況的資深管理人員、架構人員以及利害關人進行分類。這種管理才可配合整體商業風險，優先排序專案檢閱。

檢閱團隊在設計檢閱時，應配合專案團隊一起作業，以收集足夠的資訊瞭解易受攻擊面、將專案特有安全需求配對至設計元素，以及驗證模組介面的安全機制。

架構檢閱

AR 3

要求評估與驗證成品，以更詳細地瞭解保護機制。

活動

開發敏感資源的資料流程圖

根據軟體專案的商業功能進行分析，辨識高風險功能相關的系統行為細節。一般而言，高風險功能與實施敏感資料的建立、存取、更新以及刪除的動作具有相互關係。除了資料之外，高風險功能也包括了本質具有重要性，來自 Denial of Service 或洩漏方面的專案特有商業邏輯。

針對每項已辨識的資料來源或商業功能，選取並使用標準化附註以擷取相關軟體模組、資料來源、動作項目，以及傳遞於其間的訊息。配合高層級設計圖表有助於這項工作的進行，並漸漸地完成相關細節，同時移除與敏感資源無關的元素。

配合專案所建立的資料流程圖表對圖表進行分析，以決定設計中的內部壓制點 (choke point)。一般而言，這些點都是個別的軟體模組，所處理的資料具有不同敏感層級，並可存取不同商業重要性層級的多種商業功能。

建立架構檢閱的版本發行門檻

建立一致的架構檢閱方案後，下一個要執行的步驟是在軟體開發生命週期中設定特定的點，限制專案只有在執行過架構檢閱以及檢閱、核准發現問題後才可通過程序。若要完成這個步驟，必須設定基線層級的預期，例如：含有高嚴重性發現問題的所有專案不得通過此程序，或是必須由業主接受所有其他的發現問題才可通過。

一般而言，架構檢閱應該在設計階段接近尾聲時執行，以及早偵測安全問題，但必須要在專案團隊發行軟體之前執行。

針對傳統系統或非作用中的專案建立例外程序，以讓這些專案可以繼續其作業，但是必須為每個專案明確地指派接受檢閱的時間範圍，找出現有系統中的所有隱藏弱點。例外的比例應該限制於低於所有專案的 20%。

結果

- 精密檢閱系統設計中的弱點，以促成更優異的區間劃分
- 達成組織層級警覺，判斷專案是否符合架構的基線安全預期
- 比較專案，有效率的減緩已知缺陷

其他成功指標

- 過去 6 個月內，有超過 80%的專案已更新資料流程圖表
- 過去 6 個月內，有超過 75%的專案通過了架構檢閱稽核

其他成本

- 維護資料流程圖表所產生的進行中專案管理費
- 架構檢閱稽核不合格造成專案延誤所產生的組織負荷

其他人事

- 開發人員(2 天/年)
- 架構人員(1 天/年)
- 管理人員(1 至 2 天/年)
- 業主(1 至 2 天/年)
- 安全稽核人員(2 至 3 天/年)

相關目標

- 〈防禦設計〉-3
- 〈程式碼檢閱〉-3

相關法規遵從

程式碼檢閱

〈程式碼檢閱〉功能著重於從來源程式碼層級檢視軟體，以尋找安全弱點。程式碼層級弱點的概念通常很容易瞭解，但即使是擁有良好資訊的開發人員，也很容易犯錯而讓軟體可能受到威脅。

為了進行這項功能，組織應使用少量檢查清單，並只檢查最重要的軟體模組以獲得更好的效率。但隨著組織日益精進，組織可使用自動化技術，大幅改善程式碼檢閱活動的涵蓋範圍與效力。

此功能精密的佈建形式可將程式碼檢閱更深入的整合至開發程序，讓專案團隊及早發現問題。這也讓組織獲得更佳的稽核結果，並在版本發行之前設定程式碼檢閱發現問題的預期。

	CR 1	CR 2	CR 3
目標	**適時尋找基本程式碼層級的弱點以及其他高風險安全問題。**	**透過自動化，使得開發過程中能夠更準確並有效率的檢閱程式碼。**	**規定完整程式碼檢閱程序，以發現語言層級與應用程式特有的風險。**
活動	從已知安全需求建立檢閱檢查清單 執行高風險程式碼的點檢閱	利用自動化程式碼分析工具 整合程式碼分析以及開發程序	自訂應用程式特有考量的程式碼分析 建立程式碼檢閱的版本發行門檻
結果	檢視可能造成攻擊的常見程式碼弱點 少量檢閱導致嚴重安全衝擊的程式碼錯誤 安全保證的基本程式碼層級事前審查	開發人員可一致的自我檢查程式碼層級的安全弱點 定期分析結果，以編譯每個團隊安全編碼習慣的歷程資料 利害關係人可察覺到未減緩的弱點，以支援更佳的得失抵換分析	程式碼分析結果的正確性與可應用性方面，可信賴度有所增加 達成全組織的安全編碼預期基線 專案團隊具備判斷程式碼層級安全的客觀目標

程式碼檢閱

CR 1

適時尋找基本程式碼層級的弱點以及其他高風險安全問題。

活動

從已知安全需求建立檢閱檢查清單

請從專案的已知安全需求中，衍生出少量的安全程式碼檢閱檢查清單。這些項目可以是有關功能需求安全考量的特定檢查，或是根據實施語言、平台，以及典型技術堆疊等項目的安全編碼最佳實作的檢查。因為這些變化，所以通常需要涵蓋組織內不同類型軟體開發的檢查清單集。

不論是否是從公用資源或購買資源中所建立的，如開發管理人員、架構人員、開發人員以及安全稽核人員等技術利害關係人，都應該檢閱檢查清單的效力與可行性。務必保持清單的簡短，並掌握高優先順序問題，以便手動或配合簡易搜尋工具，在程式碼中簡單直接找出這些問題。您也可以使用程式碼分析自動化工具，以達到相同的目標，但也應該自訂這些工具縮小整體安全檢查集，取得小型、效益高的檢查集，以提高掃描與檢閱程序的效率。

開發人員應該針對工作職能，接受合適的檢查清單目標簡報。

執行高風險程式碼的點檢閱

如果程式碼層級弱點發生在軟體的安全關鍵部份，就會大幅增加衝擊的規模，專案團隊應該檢閱高風險模組是否含有常見的弱點。高風險功能常見的範例包括：驗證模組、存取控制執行點、階段作業管理方案、外部介面、輸入驗證工具，以及資料解析器等。

這項分析可利用程式碼檢閱檢查清單，做為開發程序的一正常部份執行，並在進行變更時，指派模組給專案團隊成員檢閱。檢閱時也可使用安全稽核人員與自動化檢閱工具。

在高風險程式碼進行變更與檢閱的開發週期中，開發管理人員應配合來自其他專案利害關係人的資源輸入，適當地分類發現問題與優先排序矯正方式。

結果

- 檢視可能造成攻擊的常見程式碼弱點
- 少量檢閱導致嚴重安全衝擊的程式碼錯誤
- 安全保證的基本程式碼層級事前審查

成功指標

- 過去 6 個月內，有超過 80%的專案團隊接受過相關程式碼檢閱檢查清單的簡報
- 過去 6 個月內，有超過 50%的專案團隊執行了高風險程式碼的檢閱
- 超過 3.0 的李克特量表得分，肯定開發人員報告的程式碼檢閱檢查清單的效用

成本

- 建置或授權程式碼檢閱檢查清單
- 高風險程式碼的程式碼檢閱活動所產生的進行中專案管理費

人事

- 開發人員(2 至 4 天/年)
- 架構人員(1 至 2 天/年)
- 管理人員(1 至 2 天/年)
- 業主(1 天/年)

相關目標

- 〈安全需求〉-1

相關法規遵從

-

CR 2

程式碼檢閱

透過自動化，使得開發過程中能夠更準確並有效率的檢閱程式碼。

結果

- 開發人員可一致的自我檢查程式碼層級的安全弱點
- 定期分析結果，以編譯每個團隊安全編碼習慣的歷程資料
- 利害關係人可察覺到未減緩的弱點，以支援更佳的得失抵換分析

其他成功指標

- 過去 6 個月內，有超過 50%的專案進行了程式碼檢閱與利害關係人簽核
- 過去 1 月內，有超過 80%的專案已可存取自動化程式碼檢閱結果

其他成本

- 程式碼分析解決方案的研究與選擇
- 自動整合的初始成本與維護
- 自動化程式碼檢閱與抵減所產生的進行中專案管理費

其他人事

- 開發人員(1 至 2 天/年)
- 架構人員(1 天/年)
- 管理人員(1 至 2 天/年)
- 安全稽核人員(3 至 4 天/年)

相關目標

相關法規遵從

活動

利用自動化程式碼分析工具

程式碼層級的許多安全弱點都很複雜而令人難以理解，並且需要謹慎的檢查才可發現。但您可以利用許多有用的自動化解決方案，以自動分析程式碼中的任何錯誤與弱點。

您可以使用商業產品與開放源碼產品，以涵蓋經常使用的程式語言與骨架。您可使用多項因素選擇適當的程式碼分析解決方案，包括檢查的縱深與準確性、產品可用性與使用模型、可擴展性與自訂功能、對組織架構與技術堆疊的適用性等。

選擇程序中，請要求專業安全技術人員，以及開發人員、開發管理人員一起參與，並配合利害關係人檢閱整體結果。

整合程式碼分析以及開發程序

一旦選取程式碼分析解決方案之後，該分析就必須整合至開發程序中，以利專案團隊使用其功能。為了有效達到此目標，您可以設定基礎架構在建置時間自動執行掃描，或是從專案程式碼儲存機制中的程式碼進行設定。依此模式，就可儘早獲得結果，也因此可讓開發團隊在軟體發行前的過程中進行自我檢查。

舊版系統或進行中的大規模專案所共有的潛在問題是，程式碼掃描工具一般會以模組方式報告未在發行版本中更新的發現問題。如果您設定了定期執行自動掃描，就不需考量上一次掃描後已新增、移除或變更的發現問題，有效避免額外檢閱。但切勿忽略其他的結果，而且開發管理人員也應該要求安全稽核人員、利害關係人以及專案團隊的參與，以形成解決其他發現問題的具體計劃。

如果程式碼檢閱中未解決的發現問題在發行時仍未解決，就必須由專案利害關係人檢閱並核准。

程式碼檢閱

CR 3

規定完整程式碼檢閱程序，以發現語言層級與應用程式特有的風險。

活動

自訂應用程式特有考量的程式碼分析

程式碼掃描工具由內建規則知識庫驅動，可根據語言 API 與常用資源庫檢查程式碼，但對於自訂 API 與設計的瞭解卻有限，只能套用類比檢查。但是，程式碼掃描工具可以透過自訂，成為功能強大的一般分析引擎，尋找組織與專案特有的安全考量。

以自訂分析的簡易性與功能而言，雖然各工具的設定細節不同，程式碼掃描工具自訂化程序通常會指定在特定 API 與函數呼叫地點執行檢查。檢查內容可包括與內部編碼標準的符合度分析、將未檢查的感染資料傳送到自訂分析、敏感資料處理的追蹤與驗證、正確使用內部 API 等項目。

若要開始自訂掃描工具，最好從共享程式知識庫入門，這樣即可將已建立的檢查工具運用在多個專案中。若要自訂程式知識庫工具，安全稽核人員應同時檢查程式碼與高層集設計，以辨識可能使用的檢查工具，並與開發人員與利害關係人一起討論實施辦法。

建立程式碼檢閱的發行門檻

若要為全部軟體專案設定程式碼層級安全基線，應該在軟體開發生命週期中建立特定的點做為檢查點，程式碼檢閱結果必須達到最低標準才可進行發行。

首先，此標準必須可明確的達成，舉例來說，選擇一或兩項弱點類型並設定標準規定，專案只要具備對應的發現問題，就不得通過。您可以藉由增加通過檢查點的額外標準，慢慢地改善基線標準。

一般而言，程式碼檢閱檢查點應該在實施階段尾聲時進行，可是一定要在版本發行前進行。

您應該針對傳統系統或未作用專案建立例外程序，以允許這些專案繼續作業，但要明確的指定減緩發現問題的時間範圍。例外的比例應該限制於低於所有專案的 20%。

結果

- 程式碼分析結果的正確性與可應用性方面，可信賴度有所增加
- 達成全組織的安全編碼預期基線
- 專案團隊具備判斷程式碼層級安全的客觀目標

其他成功指標

- 超過 50%的專案使用了程式碼分析自訂化
- 過去 6 個月內，有超過 75%的專案通過了程式碼檢閱稽核

其他成本

- 建置與維護自訂程式碼檢閱檢查
- 程式碼檢閱稽核所產生的進行中專案管理費
- 程式碼檢閱稽核不合格造成專案延誤所產生的組織負荷

其他人事

- 架構人員(1 天/年)
- 開發人員(1 天/年)
- 安全稽核人員(1 至 2 天/年)
- 業主(1 天/年)
- 管理人員(1 天/年)

相關目標

- 〈標準與法規遵從〉-2
- 〈防禦設計〉-3

相關法規遵從

安全測試

〈安全測試〉功能著重於軟體檢查，同時在執行階段環境進行作業，以尋找安全問題。這些測試活動會在預期執行環境下接受檢查，以支持軟體的保證案例，因此可看見商業邏輯內難以發現的作業不當配置或錯誤。

組織一開始可根據軟體的基本功能設定高層級測試案例，漸漸地實施安全測試自動化，以涵蓋可能在系統展現弱點的各種測試案例。

此功能佈建具有進階形式，包括可自訂自動化測試，以建立可詳細涵蓋應用程式特有考量的一套安全測試。安全測試若加上組織層級的額外可見度，將可讓組織在核准專案發行之前，設定安全測試結果的最低預期。

	ST 1	ST 2	ST 3
目標	**啟用測試程序，根據軟體需求執行基本安全測試。**	**透過自動化，使得開發過程中能夠進行更完整並有效率的安全測試。**	**要求應用程式特有的安全測試，在部署之前確定基線安全。**
活動	• 從已知安全需求衍生測試案例 • 明確評估已知安全測試案例	• 利用自動化安全測試工具 • 整合安全測試至開發程序	• 使用應用程式特有安全測試自動化 • 建立安全測試版本發行門檻
結果	📖對重要商業功能相關的安全機制期望值進行獨立驗證 📖對安全測試進行高層級事前審查 📖各軟體專案的安全測試套件特定成長	📖對軟體功能的安全進行更深入、一致的驗證 📖開發團隊在可版本發行之前自我檢查並更正問題 📖利害關係人做出風險承受度決策時，可更加警覺開放式弱點	📖全組織具備對攻擊的預期應用程式效能基線 📖自訂安全測試套件以改善自動化分析的準確度 📖專案團隊對攻擊抵抗性客觀目標的認知

安全測試

ST 1

啟用測試程序，根據軟體需求執行基本安全測試。

活動

從已知安全需求衍生測試案例

從專案的已知安全需求中辨識測試案例集，以檢查軟體功能是否正確。一般而言，這些測試案例是從有關功能需求與系統商業邏輯的安全考量衍生而來，但也應該根據實施語言或技術堆疊，將常見弱點的一般測試含括在內。

通常，您可以有效的利用專案團隊時間建立應用程式特有測試案例，並使用公用資源或購買的知識庫，以選擇安全的可應用一般測試案例。雖然不須使用安全測試工具，您仍可以使用這些工具來涵蓋一般安全測試案例。

這項測試案例計劃應該在需求和/或設計階段時執行，但一定得在發行之前進行。您應該檢閱預備使用的測試案例，依據相關開發、安全以及品保人員等因素，確定案例的可應用性、效力以及可行性。

明確評估已知安全測試案例

若針對每個專案使用已辨識的安全測試案例集，應進行測試以評估各案例的系統效能。這項作業通常要在發表之前的測試階段進行。

安全測試案例一經指定後，就可由安全專業品保人員或開發人員執行，但專案團隊初次執行安全測試案例時，應該由安全稽核人員監督，以協助並指導團隊成員。

利害關係人必須在發行或佈署之前檢閱安全測試結果，並在發表時間駁回安全測試，以表示接受所指出的風險。在後述情況中，應該建立具體的時間表以解決時間上的差異。

結果

- 對重要商業功能相關的安全機制期望值進行獨立驗證
- 對安全測試進行高層級事前審查
- 各軟體專案的安全測試套件特定成長

成功指標

- 過去 12 個月內，有超過 50%的專案指定了安全測試案例
- 過去 6 個月內，有超過 50%的利害關係人接受過專案安全測試狀態的簡報

成本

- 建置或授權安全測試案例
- 維護與評估安全測試案例所產生的進行中專案管理費

人事

- QA 測試人員(1 至 2 天/年)
- 安全稽核人員(1 至 2 天/年)
- 開發人員(1 天/年)
- 架構人員(1 天/年)
- 業主(1 天/年)

相關目標

- 〈安全需求〉-1

相關法規遵從

ST 2

安全測試

透過自動化，使得開發過程中能夠進行更完整並有效率的安全測試。

結果

- 對軟體功能的安全進行更深入、一致的驗證
- 開發團隊在可版本發行之前自我檢查並更正問題
- 利害關係人做出風險承受度決策時，可更加警覺開放式弱點

其他成功指標

- 過去 6 個月內，有超過 50%的專案進行了安全測試與利害關係人簽核
- 過去 1 年中，有超過 80%的專案能夠存取自動化安全測試結果

其他成本

- 自動化安全測試解決方案的研究與選擇
- 自動整合的初始成本與維護
- 自動化安全測試與抵減所產生的進行中專案管理費

其他人事

- 開發人員(1 天/年)
- 架構人員(1 天/年)
- 管理人員(1 至 2 天/年)
- 安全稽核人員(2 天/年)
- QA 測試人員(3 至 4 天/年)

相關目標

相關法規遵從

活動

利用自動化安全測試工具

為了測試安全問題，很可能必須針對每項軟體介面檢查大量的輸入案例，而使手動測試案例實施與執行的有效安全測試效能大幅減弱。因此，您應該使用自動化安全測試工具自動測試軟體，達到更有效率的安全測試與較高品質的結果。

您可以使用商用產品或開放源碼產品，也應該檢閱這些產品對組織的適用性。要選取合適的工具，必須以下列因素為依據：內建安全測試案例的強度與準確度、測試組織重視的測試架構類型的效力、自訂變更或新增測試案例、發現問題對組織內開發人員的品質與可使用性以及其他因素。

選擇程序中，請要求專業安全技術人員，以及開發人員、品保人員一起參與，並配合利害關係人檢閱整體結果。

整合安全測試至開發程序

組織內的專案使用工具執行自動化安全測試時，應該在開發過程中定期執行安全測試並檢閱結果。為了擴充此測試作業又不讓負擔過重，您應該配置安全測試工具定期（例如；每晚或每週）自動執行，並檢查產生的發現問題。

在需求或設計早期階段進行安全測試，將帶來很大的助益。這種以往針對功能測試案例執行的測試驅動開發方式，包含了在開發週期早期（通常在設計階段中）進行辨識並執行相關的安全測試案例。專案在自動執行安全測試案例後，若未通過測試的功能都是不存在的功能，就可進入實施階段。而必須通過所有測試之後，才算完成實施階段。這在開發週期初期就為開發人員提供了清楚、直接的目標，也可減少因為要趕上專案期限所造成的安全疑慮或被迫承受風險，導致延期發行的風險。

每次專案發行產生的自動化與手動安全測試結果，應該上呈給管理人員與商業利害關係人進行檢閱。如果發行中仍有未解決問題為已核准風險，利害關係人與開發管理人員應該合作建立解決問題的具體時間表。

安全測試

ST 3

要求應用程式特有的安全測試，在部署之前確定基線安全。

活動

使用應用程式特有安全測試自動化

不論是透過自訂安全測試工具、增強一般測試案例執行工具，或是建置自訂測試工具，專案團隊都應該正式實施安全需求並建立自動化檢查工具集，以測試已實施商業邏輯的安全。

另外，若可自訂自動化安全測試工具，以進一步瞭解所測試專案中特定軟體介面的繫結，將可大幅改善許多資動畫安全測試工具的準確度與涵蓋縱深。來自法規遵從性或技術標準的組織特有考量，可更進一步的編碼為可重複使用的集中測試組，以簡化稽核資料收集與每個專案的管理可視性。

專案團隊應該根據軟體的商業功能，著重在精細度安全測試案例的建置，而由安全稽核人員帶領的組織層級團隊，應該著重於法規遵從與內部標準的自動測試規格。

建立安全測試版本發行門檻

若要避免發行軟體含有顯而易見的安全錯誤，必須將軟體開發生命週期中的特定點視為檢查點，而已建立的安全測試案例集必須通過此檢查點才可發表專案。這樣就可建立出所有專案都必須通過的安全測試基準。

因為一開始就加入太多的測試案例會造成額外負荷成本過大，您可以在開始時選擇一項或兩項安全問題，並將各種測試案例內入各問題，預期只要專案無法通過其中一項測試就算不合格。接著慢慢選取額外安全問題並加入各種對應測試案例，就可逐漸改善此基線。

一般而言，安全測試檢查點應該在實施或測試階段尾聲時進行，可是一定要在發行前進行。

您應該針對傳統系統或未作用專案建立例外程序，以允許這些專案繼續作業，但要明確的指定減緩發現問題的時間範圍。例外的比例應該限制於低於所有專案的 20%。

結果

- 全組織具備對攻擊的預期應用程式效能基線
- 自訂安全測試套件以改善自動化分析的準確度
- 專案團隊對攻擊抵抗性客觀目標的認知

其他成功指標

- 超過 50%的專案使用了安全測試自訂化
- 過去 6 個月內，有超過 75%的專案通過所有安全測試

其他成本

- 建置與維護安全測試自動化自訂
- 安全測試稽核程序所產生的進行中專案管理費
- 安全測試稽核不合格造成專案延誤所產生的組織負荷

其他人事

- 架構人員(1 天/年)
- 開發人員(1 天/年)
- 安全稽核人員(1 至 2 天/年)
- QA 測試人員(1 至 2 天/年)
- 業主(1 天/年)
- 管理人員(1 天/年)

相關目標

- 〈標準與法規遵從〉-2
- 〈防禦設計〉-3

相關法規遵從

弱點管理

〈弱點管理〉功能著重在處理弱點報告與操作意外的組織內程序。這些事件通常導致混亂與無資訊的回應，若建立起程序，組織中的專案就能具有一致的預期並提高回應效率。

組織一開始遇上意外時，可先指派少量的角色，再逐漸發展出較正式的意外回應程序，以確保可察覺意外發生並加以追蹤。溝通方式也會有所改善，以增加對程序的整體瞭解。

弱點管理的進階形式包括徹底解析意外與弱點報告，以收集詳細指標做為組織下游行為的回應。

	VM 1	VM 2	VM 3
目標	**瞭解對弱點報告或意外的高層級回應計畫。**	**詳細規劃預期回應程序，以改善一致性與溝通表達力。**	**改善回應程序內的分析與資料收集，作為主動規劃的回應參考。**
活動	辨識安全問題的連絡點 建立非正式安全回應團隊	建立一致的意外回應程序 採用安全問題揭露程序	進行意外的根本原因分析 收集每件意外的指標
結果	建立少量程序以處理高優先順序弱點或意外 建立利害關係人通告與報告安全衝擊事件的骨架 進行安全問題處理的高層級事前審查	建立處理第三方弱點報告的溝通計劃 讓軟體操作者獲得發表安全修補程式的清楚程序 意外相關追蹤、處理以及內部溝通的正式程序	每件意外後，對組織改善的詳細回應 弱點與洩漏的大略成本預估 利害關係人可根據歷程意外趨勢，做出更佳的得失抵換決策

弱點管理

VM 1

瞭解對弱點報告或意外的高層級回應計畫。

活動

辨識安全問題的連絡點

針對組織內每個部門或每個專案團隊，建立連絡點以做為安全資訊的溝通樞紐。一般而言，此責任並不須花費個人太多的時間，這麼做目的是要新增弱點管理的架構與管理。

發生意外而必須使用連絡點的範例有：接受來自外部實體的弱點報告、使用中軟體的洩漏或其他安全失敗、內部發現高風險弱點等等。在事件發生時、最近的連絡就可做為受影響專案團隊的額外資源與顧問，除了提供技術指導方針，也要對其他利害關係人簡報工作進度受影響的情況。

請從組織內對軟體專案有廣泛知識的專業安全技術人員或管理人員，選出連絡點。至少每 6 個月，要對指派的安全連絡點清單進行集中維護與更新。另外，請對組織內員工發佈或通告此清單，以讓員工要求援助並直接協同處理安全問題。

建立非正式安全回應團隊

請從被賦予安全連絡點責任的個人清單，或是專用安全人員中，選出一個小組以做為中央技術安全回應團隊。該團隊的責任包括直接掌控安全意外或弱點報告，並負責予以分類並減緩，以及對利害關係人報告。

依據安全回應團隊成員啟動時的責任，成員也必須在意外發生時負責簡報執行狀況與向上溝通。安全回應團隊大部分的時間可能無法達成這樣的產能，但他們必須具備彈性以快速回應問題，或是有一套流暢程序以將意外指派給另一位團隊成員。

回應團隊至少應該每年舉行一次會議，以簡報回應程序的安全連絡點，以及來自專案團隊有關安全相關報告的高層級期望。

結果

- 建立少量程序以處理高優先順序弱點或意外
- 建立利害關係人通告與報告安全衝擊事件的骨架
- 進行安全問題處理的高層級事前審查

成功指標

- 過去 6 個月內，有超過 50%的組織接受過最近安全連絡點的簡報
- 過去 12 個月內，已舉行過 1 次以上的安全回應團隊

成本

- 員工擔任安全連絡點角色所造成的持續變化專案負荷
- 辨識安全回應團隊的適用性

人事

- 安全稽核人員(1 天/年)
- 架構人員(1 天/年)
- 管理人員(1 天/年)
- 業主(1 天/年)

相關目標

- 〈學習與指導方針〉-2
- 〈策略規劃〉-3

相關法規遵從

VM 2

弱點管理

詳細規劃預期回應程序，以改善一致性與溝通表達力。

結果

- 建立處理第三方弱點報告的溝通計劃
- 讓軟體操作者獲得發表安全修補程式的清楚程序
- 意外相關追蹤、處理以及內部溝通的正式程序

其他成功指標

- 過去 6 個月內，有超過 80%的專案團隊接受過意外回應程序的簡報
- 過去 6 個月內，有超過 80%的利害關係人接受了安全問題揭露簡報

其他成本

- 意外回應程序的持續組織負荷

其他人事

- 安全稽核人員(3 至 5 天/年)
- 管理人員(1 至 2 天/年)
- 業主(1 至 2 天/年)
- 支援/操作人員(1 至 2 天/年)

相關目標

相關法規遵從

活動

建立一致的意外回應程序

從非正式安全回應團隊所延伸出的任務為，明確地記錄組織的意外回應程序，以及團隊成員應該遵守的程序。另外，安全回應團隊的每位成員至少每年要接受一次有關此資料的訓練。

健全的意外回應程序有數個原則，包括初始分類以避免額外損害、變更管理與修補應用程式、管理專案人事與其他牽涉的人員、收集與保存意外發生的證據、限制透露給利害關係人的意外相關溝通、定義良好的利害關係人報告與/或溝通樹狀結構等原則。

安全回應人員應該與開發團隊合作進行技術分析，以驗證有關每件意外或弱點報告的事實與假設。相同的，當專案團隊偵測到意外或高風險弱點時，應該按照內部程序以連絡安全回應團隊的成員。

採用安全問題揭露程序

對於大多數組織來說，發生安全問題的消息最好不要走漏而變得眾所皆知，但是得先做到數個安全問題由內向外溝通的重要方式。

第一個也是最常見的方法是，透過對組織製作的軟體建立並部署安全修補程式。一般而言，如果所有的軟體專案只在內部使用，問題比較不嚴重，但對於軟體在組織外部操作的全部條件而言，一定要有修補程式的發行程序。發表程序應包含數個因素，包括發 行修補程式前變更管理與回歸測試、針對修補程式配合指派的嚴重性分類以通告操作者/使用者、分散技術細節而無法直接產生惡意探索等因素。

另一個與外部溝通的方法是，透過提報組織軟體有安全弱點的第三方。採用並對外公佈附有回應時間點的預期程序，可促使弱點報告人員遵守責任揭露 (responsible disclosure) 實作。

最後，若意外涉有個人身份資訊與其他敏感資料類型的竊取行為，則依許多國家的法律，都必須進行外部溝通。如果發生這種意外，安全回應團隊應該與管理人員以及商業利害關係人合作，決定適當的後續步驟。

弱點管理

VM 3

改善回應程序內的分析與資料收集，作為主動規劃的回應參考。

活動

進行意外的根本原因分析

意外回應程序可能會花費許多時間，但仍應擴展其範圍以包括額外的分析以辨識關鍵、潛在的安全失敗。這些根本原因可能為技術問題（例如；程式碼層級弱點、配置錯誤等），或是人為/程序問題（例如；社交工程、未遵守程序等）。

只要辨識出意外的根本原因，就應該以此原因做為工具，尋找組織內可能發生類比意外的其他潛在弱點。原始意外回應工作結案過程中，必須針對主動減緩各項已辨識弱點的其他建議，進行溝通。

以根本原因分析為根據的所有建議，都應該交由管理與相關商業利害關係人檢閱，以便排程減緩活動或記下已接受風險。

收集每件意外的指標

組織可藉由集中程序處理所有洩露與高優先順序弱點報告，長時間衡量趨勢，以判斷安全保證開發案的衝擊與效率。

過往意外的記錄應該至少每 6 個月儲存與檢閱一次。將類似的意外設為群組，並簡易計算每種問題類型的總次數。對意外其他的衡量指標包括軟體專案受意外影響的頻率、系統停機時間與停用成本、處理與清除意外所需花費的人力資源、長程成本預估（例如；法規罰金或品牌傷害）等。若根本原因的本質是技術問題，則辨識哪一類型的主動檢閱或作業實作可能及早偵測或降低損害，也是相當有助益的。

此資訊可將意見回應具象化為方案規劃程序，因為這代表組織長時間以來所感受到的實際安全衝擊。

結果

- 每件意外後，對組織改善的詳細回應
- 弱點與洩漏的大略成本預估
- 利害關係人可根據歷程意外趨勢，做出更佳的得失抵換決策

其他成功指標

- 過去 6 個月內，有超過 80%的意外已記錄了根本原因與更進一步的建議
- 過去 6 個月內，有超過 80%的意外已進行指標校正

其他成本

- 進行意外深入研究與分析所造成的持續組織負荷
- 收集與檢閱意外指標所造成的持續組織負荷

其他人事

- 安全稽核人員(3 天/年)
- 管理人員(2 天/年)
- 業主(2 天/年)

相關目標

相關法規遵從

基礎架構強化

〈基礎架構強化〉的功能著重在，增強提供組織軟體主機服務的執行階段環境。因為軟體的安全作業會受到外部元件問題的毀壞，直接強化主要基礎架構可改善組織軟體的整體安全情勢。

一開始可藉由簡易的追蹤與分配有關作業環境的資訊，讓開發團隊獲得較完善的資訊，而組織會改進採用可度量方式以管理安全修補程式的部署，並為作業環境配備早期預警偵測程式，在造成損害前偵測潛在安全問題。

隨著組織逐漸改善，作業環境可藉由部署保護工具獲得更進一步的檢閱與強化，並增加多層防護與安全網，在發生惡意探索弱點的狀況時限制損傷。

	IH 1	IH 2	IH 3
目標	**瞭解應用程式與軟體元件的基線作業環境。**	**藉由強化作業環境以改善應用程式作業的可信賴程度。**	**對已知最佳實作驗證應用程式的運作狀況與作業環境的狀態。**
活動	維護作業環境規格 辨識並安裝重要安全修補程式	建立基礎架構修補程式管理程序 監視基線基礎架構配置狀態	辨識並部署相關作業保護工具 對基礎架構配置展開稽核方案
結果	更加瞭解開發團隊內的作業預期 已根據良好的時間表，對來自潛在基礎架構的高優先順序風險進行減緩 軟體操作者對基礎架構的重要安全維護具有高層級計劃	精細驗證作業中的系統安全特性 對基礎架構風險減緩的正式預期時間表 利害關係人對軟體專案的目前作業狀態具有一致性的體認	搭配安全分層檢查增強作業環境 已建立並衡量作業維護與效能的目標 透過外部相依性缺陷降低成功攻擊的可能性

基礎架構強化

IH 1

瞭解應用程式與軟體元件的基線作業環境。

活動

維護作業環境規格

應該針對每個專案，建立與維護預期作業平台的具體定義。此規格應該根據組織，決定是否要與開發人員、利害關係人、支援與作業群組等一起建立。

此規格的一開始，應該根據軟體商業功能，擷取有關作業環境的全部正確細節。這些細節包括了程序工具架構、作業系統版本、必要軟體、衝突軟體等因素。另外，請記下所有影響軟體行為的作業環境使用者或操作者已知可配置選項。

另外，辨識在專案設計與實施階段所做的所有作業環境相關假設，並將假設擷取至規格。

如果軟體設計或預期作業環境進行變更，則應該針對作用中專案，至少每 6 個月（或更頻繁）對此規格進行檢閱與更新。

辨識並安裝重要安全修補程式

大多數應用程式都是在另一個大規模軟體堆疊上執行的軟體，此堆疊包括了內建程式語言資源庫、第三方元件與開發骨架、基礎作業系統等。因為該大規模軟體堆疊中所有模組含有的安全缺陷，會影響組織軟體的整體安全，所以必須為技術堆疊安裝重要的安全更新。

因此，您必須執行定期的研究或持續觀察高風險相依性，以得知安全缺陷最新的修復程式。一旦辨識出會影響軟體專案安全情勢的重要修補程式後，就應該計劃讓受影響的使用者與操作者更新他們的安裝軟體。進行此更新的詳細辦法，會根據軟體專案類型而有所不同。

結果

- 更加瞭解開發團隊內的作業預期
- 已根據良好的時間表，對來自潛在基礎架構的高優先順序風險進行減緩
- 軟體操作者對基礎架構的重要安全維護具有高層級計劃

成功指標

- 過去 6 個月內，有超過 50%的專案已更新作業環境規格
- 過去 6 個月內，有超過 50%的專案已更新相關重要安全修補程式的清單

成本

- 建置與維護作業環境規格所產生的進行中專案管理費
- 監視與安裝重要安全更新所產生的進行中專案管理費

人事

- 開發人員(1 至 2 天/年)
- 架構人員(1 至 2 天/年)
- 管理人員(2 至 4 天/年)
- 支援/操作人員(3 至 4 天/年)

相關目標

- 〈作業能力〉-2

相關法規遵從

IH 2

基礎架構強化

藉由強化作業環境以改善應用程式作業的可信賴程度。

結果

- 精細驗證作業中的系統安全特性
- 對基礎架構風險減緩的正式預期時間表
- 利害關係人對軟體專案的目前作業狀態具有一致性的體認

其他成功指標

- 過去 12 個月內，有超過 80%的專案團隊接受過修補程式管理程序的簡報
- 過去 6 個月內，有超過 80%的利害關係人瞭解目前的修補程式狀態

其他成本

- 修補程式管理與監視所造成的持續組織負荷
- 建置或授權基礎架構監視工具

其他人事

- 架構人員(1 至 2 天/年)
- 開發人員(1 至 2 天/年)
- 業主(1 至 2 天/年)
- 管理人員(1 至 2 天/年)
- 支援/操作人員(3 至 4 天/年)

相關目標

相關法規遵從

活動

建立基礎架構修補程式管理程序

組織內部應該建立一項持續的程序，捨棄重要修補程式特定應用方式而採用較正式的程序，以在套用安全修補程式至作業環境時取得一致性。

此程序應以最基本的形式呈現，以確保可呈現出軟體版本發行與安全修補程式應用之間的發展情形。為了提高這個程序的效率，組織一般會讓較低優先順序的修補程式擁有高度延遲，例如：重要修補程式最多 2 天要檢查一次，而低優先順序的修補程式則可最多 30 天再檢查。

此活動主要由支援與作業員工執行，但也應該定期與開發部門舉行會議，以讓整件專案獲得過往變更並排程升級。

另外，開發人員應該共享軟體專案內部相依的第三方元件清單，讓支援與作業員工得以監看這些元件，並提示開發團隊何時需要進行升級。

監視基線基礎架構配置狀態

監視與管理組成軟體專案基礎架構的各種元件修補程式，是一項複雜的作業，因此應該使用自動化工具，自動監視系統的配置完整性。

您可以使用商用或開放源碼工具提供這類功能，專案團隊應該根據組織需求的適用性選擇解決方案。一般的選擇標準包括部署與自訂的簡易性、對組織平台與技術堆疊的可應用性、變更管理與警示的內建功能、指標收集與趨勢追蹤等。

除了主機與平台檢查外，應該自訂監視自動化，以執行應用程式特有運作狀況檢查與配置驗證。支援與作業人員應該與架構人員、開發人員合作，以決定特定軟體專案的最佳監視量。

最後，部署監視基礎架構配置狀態的解決方案後，應該收集未預期警示或配置變更，並定期由利害關係人每週（至少每季一次）檢閱一次。

基礎架構強化

IH 3

對已知最佳實作驗證應用程式的運作狀況與作業環境的狀態。

活動

辨識並部署相關作業保護工具

為了替軟體在其作業環境中建置較好的保證案例，您可以使用額外的工具加強整體系統的安全情勢。作業環境的變化很劇烈，因此必須依照專案內容，考量給予保護的技術適用性。

常用的保護工具包括網路應用程式防火牆、網路服務的 XML 安全閘道、用戶端/內嵌系統的反破壞與混亂化封包、網路入侵偵測/傳統基礎架構的防治系統、事發記錄彙總工具、以主機為基礎的整體驗證工具等。

根據組織與專案特有知識，技術利害關係人應該與支援及作業員工合作，以辨識並對商業利害關係人建議選取的作業保護工具。若依據降低風險與實施成本考量，某工具被利害關係人視為有利的投資，關係人也應該同意進行試驗、廣泛執行以及持續維護等計劃。

對基礎架構配置展開稽核方案

進行定期專案層級稽核時，請擴展檢閱範圍以包括與強化作業環境有關的成品檢視動作。除了作業環境的最新規格之外，稽核也應該檢查目前的修補程式狀態與自上一次稽核之後的歷程資料。藉由點選進入監視工具，稽核也可驗證有關應用程式配置管理與歷程變更的關鍵因素。稽核也應該檢查針對軟體架構類型可用的工具，檢查作業保護工具的使用。

專案初始發行與部署之後，可以隨時進行基礎架構的稽核，但至少應該每 6 個月執行一次。至於目前未作用中的舊系統或專案，基礎架構稽核仍應該由商業利害關係人執行與檢閱。您可建立例外程序，以允許這些專案繼續作業，但要明確的指定減緩所發現問題的時間範圍。例外的比例應該限制於低於所有專案的 20%。

結果

- 搭配安全分層檢查增強作業環境
- 已建立並衡量作業維護與效能的目標
- 透過外部相依性缺陷降低成功攻擊的可能性

其他成功指標

- 過去 6 個月內，有超過 80%的利害關係人接受過相關作業保護工具的簡報
- 過去 6 個月內，有超過 75%的專案通過了基礎架構稽核

其他成本

- 研究與選擇作業保護解決方案
- 建置或授權作業保護工具
- 維護保護工具所造成的持續作業負荷
- 基礎架構相關稽核所產生的進行中專案管理費

其他人事

- 業主(1 天/年)
- 管理人員(1 至 2 天/年)
- 支援/操作人員(3 至 4 天)

相關目標

- 〈標準與法規遵從〉-2

相關法規遵從

-

作業能力

〈作業能力〉功能著重於收集專案團隊建置軟體的重要安全資訊，並將資訊傳達給軟體使用者與操作者。缺少這項資訊，即使是設計最安全的軟體也帶有潛藏的風險，因為無法在部署地點得知重要安全特性與選擇。

一開始以少量文件進行，擷取對使用者與操作者衝擊最大的細節，接著組織就可逐漸建立起完整作業安全指南，並隨著各版本發行一起發佈。

作業能力具有進階形式，可對個別專案團隊進行組織層級檢查，以確保根據預期擷取並共享資訊。

	OE 1	OE 2	OE 3
目標	**讓開發小組與操作者能夠針對重要的安全相關資料進行討論。**	**透過詳細程序的佈建，提昇連續安全作業的預期結果。**	**規定安全資訊的溝通，並驗證成品的完整性。**
活動	擷取部署的重要安全資訊 記載典型應用程式警示的程序	建立各版本發行的變更管理程序 維護正式作業安全指南	擴展作業資訊的稽核方案 執行應用程式元件的程式碼簽署
結果	由於對正確作業具備更妥善的瞭解，使軟體安全情勢獲得特定改善 操作者與使用者能夠認知自己在確保安全部署工作中的角色 改善了軟體開發者與使用者之間的重要安全資訊溝通	軟體版本發行時附有安全相關變更的詳細指導方針 資訊儲存區中，各應用程式的安全作業程序已獲得變更 調整開發人員、操作者以及使用者之間的作業預期	對安全相關文件的預期建立起全組織的理解 利害關係人更能根據部署與作業的意見回應，做出更佳的得失抵換決策 操作者/使用者可獨立驗證軟體版本發行的完整性

作業能力

OE 1

讓開發小組與操作者能夠針對重要的安全相關資料進行討論。

活動

擷取部署的重要安全資訊

專案團隊擁有軟體特有知識後，應該可辨識安全相關配置與作業資訊，以傳達給使用者與操作者。這可讓軟體在部署地點的實際安全情勢，與專案團隊的設計目的相符。

這項分析應該從架構人員與開發人員開始，建立一份內建於軟體的安全功能清單。此清單也應該擷取有關配置選項與其安全衝擊的資訊。對於提供數種不同部署模型的專案，應該記下每項模型的安全影響，以讓使用者與操作者更加瞭解他們的選擇會造成什麼衝擊。

整體而言，清單應該簡短並擷取最重要的資訊。清單一經建立後，就應該由專案團隊與商業利害關係人檢閱並同意。另外，配合少數操作者或使用者一起檢閱此清單也可提升效率，並確保資訊易於理解與執行。專案團隊可配合每次發行檢閱並更新此資訊，但必須至少每 6 個月執行一次。

記載典型應用程式警示的程序

專案團隊擁有軟體行為的特定知識後，應該可辨識出最重要的錯誤，並提示需要使用者/操作者注意的訊息。針對每項辨識出的事件，應擷取資訊讓使用者/操作者瞭解回應事件所需的適當動作。

從軟體可能產生的大規模事件集中，根據軟體商業目的的相關性，選取最高優先順序的事件集。這應該會包括所有安全相關事件，但也可能包括與軟體運作狀況以及配置狀態的相關重要錯誤與警示。

針對每項事件，應該擷取執行建議以通知使用者與操作者，必要的接續步驟以及事件的潛在根本原因。專案團隊必須檢閱這些程序，並每 6 個月更新一次，但更新的頻率也可比較密集（例如：每次版本發行時）。

結果

- 由於對正確作業具備更妥善的瞭解，使軟體安全情勢獲得特定改善
- 操作者與使用者能夠認知自己在確保安全部署工作中的角色
- 改善了軟體開發者與使用者之間的重要安全資訊溝通

成功指標

- 過去 6 個月內，有超過 50%的專案已更新部署安全資訊
- 過去 6 個月內，有超過 50%的專案已更新事業作業程序

成本

- 維護安全資訊部署所產生的進行中專案管理費
- 維護重要作業程序所產生的進行中專案管理費

人事

- 開發人員(1 至 2 天/年)
- 架構人員(1 至 2 天/年)
- 管理人員(1 天/年)
- 支援/操作人員(1 天/年)

相關目標

相關法規遵從

OE 2

作業能力

透過詳細程序的佈建，提昇連續安全作業的預期結果。

結果

- 軟體版本發行時附有安全相關變更的詳細指導方針
- 資訊儲存區中，各應用程式的安全作業程序已獲得變更
- 調整開發人員、操作者以及使用者之間的作業預期

其他成功指標

- 過去 6 個月內，有超過 50%的專案已更新變更管理程序
- 過去 6 個月內，有超過 80%的利害關係人接受過作業安全指南的簡報

其他成本

- 維護變更管理程序所產生的進行中專案管理費
- 維護作業安全指南所產生的進行中專案管理費

其他人事

- 開發人員(1 至 2 天/年)
- 架構人員(1 至 2 天/年)
- 管理人員(1 天/年)
- 支援/操作人員(1 天/年)

相關目標

- 〈基礎架構強化〉-1

相關法規遵從

-

活動

建立各版本發行的變更管理程序

為了正式讓使用者與操作者知道軟體內相關變更的更新情況，各版本發行必須包括與升級以及初次安裝有關的變更管理程序。整體而言，目標是擷取預期的附加步驟，以確保部署成功，不會造成多餘停機時間或降低安全情勢。

若要在開發階段中建立這些程序，專案團隊應該設定少量的內部程序，擷取會衝擊部署的相關項目。在開發週期早期建立此程序可大幅提高效率，如此在需求、設計以及實施階段時，只要辨識出這種資訊，就可予以保留。

每次版本發行前，專案團隊都應該檢閱所有清單，確認完整性與可實行性。對某些專案而言，隨著版本發行產生的延伸變更程序可能必須進行特殊處理，例如：建置自動化升級指令集，避免部署時出錯。

維護正式作業安全指南

專案團隊應可從重要軟體事件所擷取的資訊以及處理每項事件的程序，開始建立並維護正式指南，將使用者與操作者必須知道的所有安全相關資訊納入其中。

一開始，此指南應該建立於系統的已知資訊，例如：安全相關的配置選項、事件處理程序、安裝與升級指南、作業環境規格、對部署環境的安全相關假設等。再從此擴充為正式的作業安全指南，詳細說明每個項目以涵蓋更多的細節，讓大多數使用者與操作者才可以他們可能遇上的所有問題。對於大規模或複雜的系統，這項作業具有相當的難度，所以專案團隊應該與商業利害關係人合作，以決定文件的適用層級。另外，專案團隊應該記錄所有可增強安全性的部署建議。

初始安裝後，專案團隊應檢閱作業安全指南，並配合每一次版本發行進行更新。

作業能力

OE 3

規定安全資訊的溝通，並驗證成品的完整性。

活動

擴展作業資訊的稽核方案

進行定期專案層級稽核時，請擴展檢閱範圍以包括與安全作業能力有關的成品檢視動作。您應該檢查專案，確定專案具備軟體特定項目相關的最新、完整作業安全指南。

這些稽核應該在開發週期即將結束、發行軟體之前執行，但一定要完成並通過稽核才可發表軟體。至於目前未作用中的舊系統或專案，若不須再進行額外作業能力稽核，則應該進行此類型的稽核以及一次性的工作，解決發現問題並驗證稽核法規遵從。

發行之前，商業利害關係人應該先檢閱稽核結果。您可建立例外程序，允許未通過稽核的專案繼續進行發行，但這些專案應明確指定減緩所發現問題的時間範圍。例外的比例應該限制於低於進行中專案的 20%。

執行應用程式元件的程式碼簽署

雖然程式碼簽署通常搭配特殊用途軟體使用，仍然可允許使用者與操作者執行軟體的完整性檢查，如此就可用加密編碼的方式驗證模組或發行的可信賴性。專案團隊藉由簽署軟體模組，讓部署的作業獲得更多保證，不致在作業環境中讓部署軟體受到任何毀損或修改。

簽署程式碼會因為組織的簽署憑證，造成管理上的負荷。組織必須遵照安全金鑰管理程序，確保簽署金鑰的持續機密性。專案利害關係人在處理任何加密編碼金鑰時，必須同時想好如何處理加密編碼相關的常見作業問題，例如：金鑰自動重建、金鑰洩漏或遺失金鑰。

因為程式碼簽署並不適用所有專案，架構人員與開發人員應與安全稽核人員以及商業利害關係人合作，判斷軟體那一部份應該進行簽署。專案慢慢開發的過程中，這份清單應配合各版本發行進行檢閱，尤其是在新增模組或變更先前簽署的元件時。

結果

- 對安全相關文件的預期建立起全組織的理解
- 利害關係人更能根據部署與作業的意見回應，做出更佳的得失抵換決策
- 操作者/使用者可獨立驗證軟體版本發行的完整性

其他成功指標

- 過去 6 個月內，有超過 80%的專案已更新作業安全指南
- 過去 6 個月內，有超過 80%的利害關係人接受過程式碼簽署選項與狀態的簡報

其他成本

- 稽核作業指南所產生的進行中專案管理費
- 管理程式碼簽署憑證所產生的進行中專案管理費
- 識別與簽署程式模組所產生的進行中專案管理費

其他人事

- 開發人員(1 天/年)
- 架構人員(1 天/年)
- 管理人員(1 天/年)
- 安全稽核人員(1 至 2 天/年)

相關目標

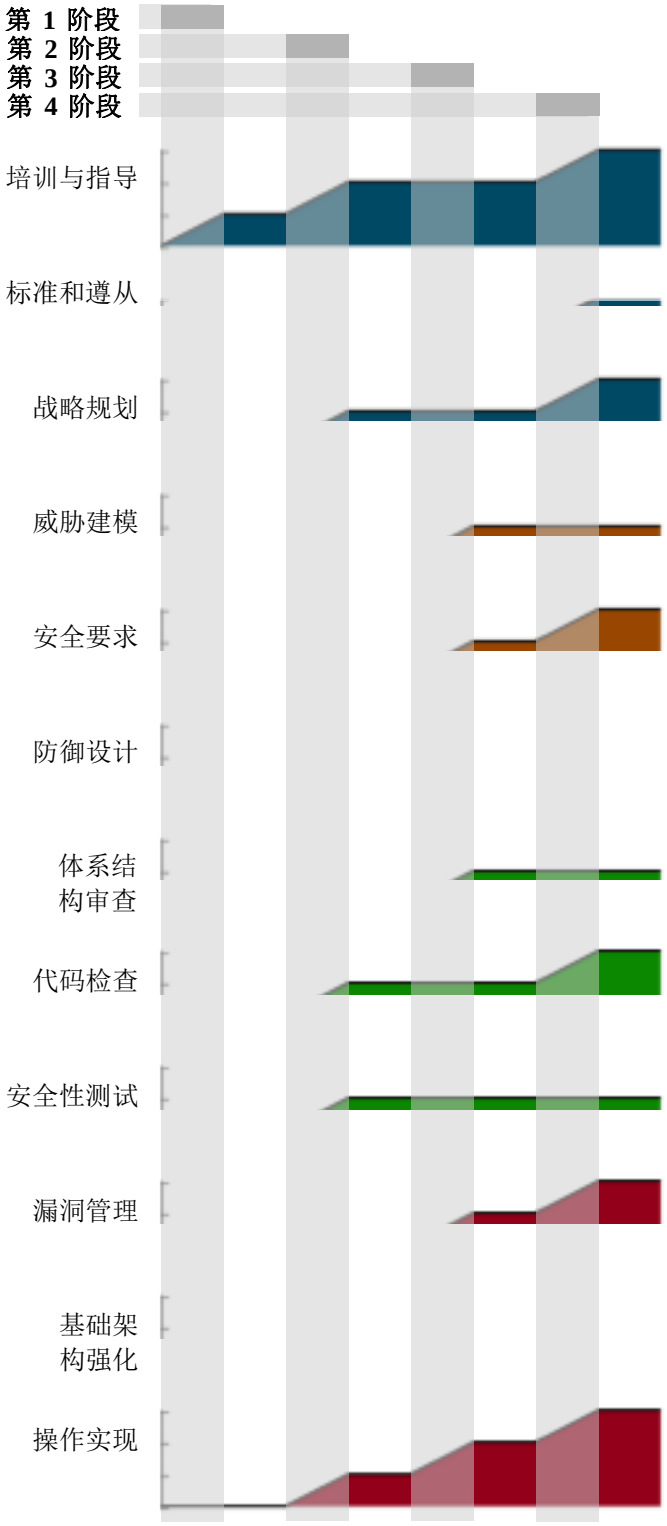
相關法規遵從

路线图

本节包含的路线图示例，详细描述了根据高级组织类型实现 SAMM 中目标的建议的顺序。建议的每个路线图都重点说明了交叉事项，以显示与建议不同的常见案例。

独立软件供应商

路线图示例



独立软件供应商 (Independent Software Vendor, ISV) 从事构建及销售软件组件和应用程序的核心业务。

由于最初目的是限制可影响客户和用户业务的常见漏洞，因而早期行动专注于代码审查和安全性测试。

在转向更主动地避免产品规格出现安全错误时，组织会逐渐增加针对安全要求的措施。

此外，为了尽量降低已发现的安全问题所造成的影响，组织会逐渐增加漏洞管理措施。

随着组织逐渐成熟，会增加“操作实现”功能的知识转移措施，以更好地为客户和用户提供软件安全操作信息。

第 1 阶段

对员工进行技术安全意识的培训
制定和维护技术指导制度
估计总体业务风险概况
EG 1 建立和维护保证方案
SP 1 根据业务功能，制定安全要求
使用指导原则、标准和标准遵从资源
SR 1 根据已知的安全要求，创建审查检查表
对高风险代码执行定点审查
CR 1 根据已知的安全要求，创建测试案例
明确评估已知的安全测试案例
ST 1 确定安全问题的联络点
创建非正式的安全响应团队
VM 1

第 2 阶段

对员工进行针对特定角色的应用程序安全培训
EG 2 聘用安全指导员，增强项目团队
根据业务风险对数据和应用程序进行分类
SP 2 建立每个分类的安全目标
建立并维护每个应用程序的攻击树
TM 1 根据软件体系结构，制定攻击者配置文件
确定软件攻击面
AR 1 根据已知的安全要求分析设计
利用自动代码分析工具
CR 2 将代码分析集成到开发流程
利用自动安全性测试工具
ST 2 将安全性测试集成到开发流程
捕获关于部署的重要安全信息
OE 1 记录典型应用程序警报的步骤

第 3 阶段

确定并监视外部标准遵从的驱动因素
SC 1 建立和维护标准遵从检查表
用应用程序特定的数据详细描述攻击树
TM 2 采用一种衡量威胁的权重系统
为每个项目建立和维护滥用案例模型
SR 2 根据已知风险，指定安全要求
维护推荐软件框架的列表
DD 1 将安全原则显式应用于外围界面
检查是否完整提供了安全机制
AR 1 为项目团队部署设计审查服务
建立一致的事件响应流程
VM 2 采用安全问题披露流程
创建每次发布的变更管理步骤
OE 2 维护正式的操作安全指南

第 4 阶段

创建应用程序安全支持门户网站
EG 3 建立基于角色的考试/认证制度
建立安全和标准遵从的内部标准
SC 2 建立项目审核实践
定期进行业界范围的成本比较
SP 3 收集历史安全费用的标准
将安全要求写入供应商协议
SR 3 扩展审核计划以满足安全需求
针对应用程序特定的问题自定义代码分析
CR 3 为代码审查建立发布关卡
分析事件的根源
VM 3 收集每一事件的衡量指标
扩展操作信息的审核程序
OE 3 对应用程序组件执行代码签名

对组织的路线图有影响的 ISV 具有一些共同的特征。

外包式开发

对于采用外部开发资源的组织，通常需要限制代码的访问权限，这会导致组织优先执行安全要求任务，而不会先执行代码审查任务。此外，如果在较早的阶段执行威胁建模，组织便可以向外包开发人员提出更明确的安全要求。由于有关软件配置的专业技术通常是外包组中最重要的，因此必须制定合同以说明与操作实现相关的措施。

连接 INTERNET 的应用程序

构建使用在线资源的应用程序的组织存在一些额外的风险，这些风险源自运行面向 Internet 系统的面向 Internet 的核心基础架构。考虑到这方面的风险，组织应该将基础架构强化措施添加到路线图中。

驱动程序与嵌入式开发

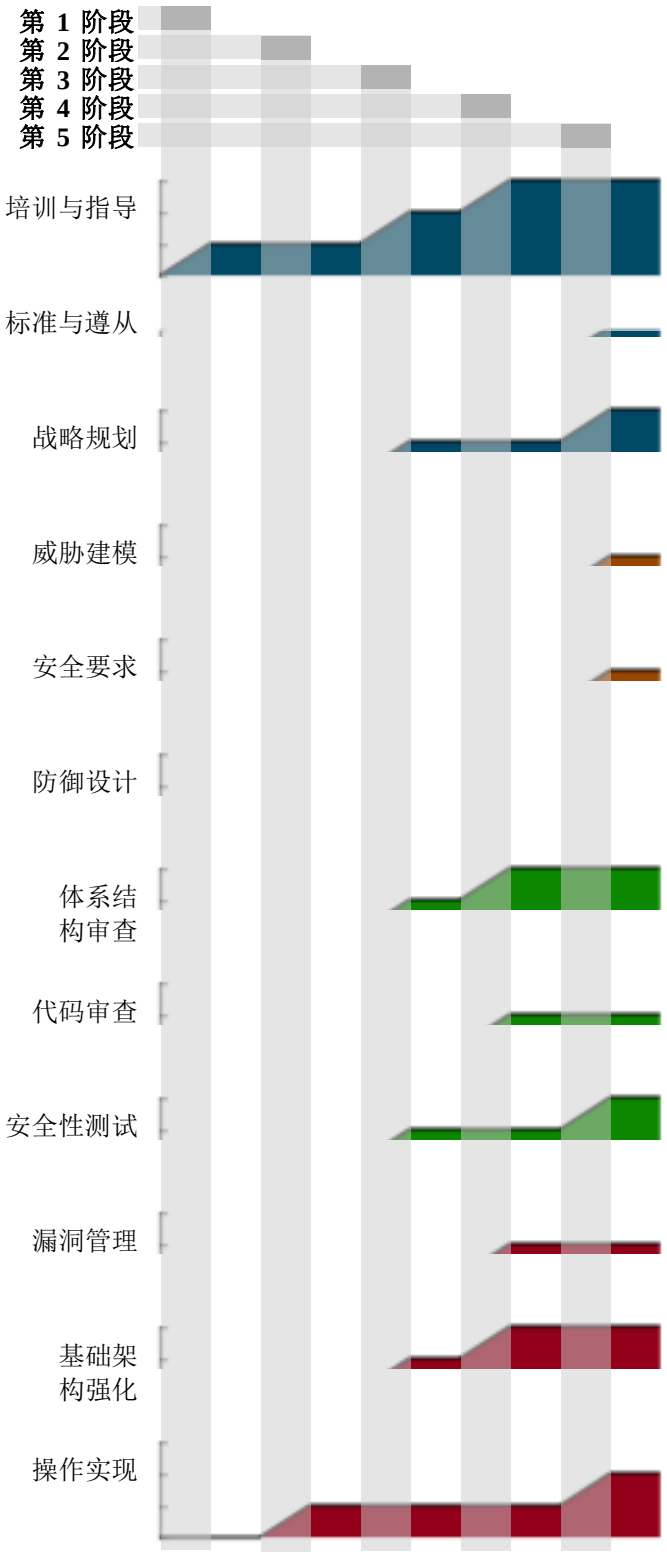
对于为嵌入式系统构建低级驱动程序或软件的组织而言，软件设计方面的安全漏洞更具破坏性，且修复成本更高。因此，必须将路线图修改为强调在早期阶段执行防御设计和体系结构审查措施。

通过收购而发展壮大的组织

在通过收购而发展壮大的组织中，通常存在遵循不同开发模式的若干个项目团队，其安全相关任务的等级也不同。对于这类组织，如果正在开发不同类型的软件，那么需要为各个部门或项目团队分别制定相应的路线图，以解决各个项目的起点和特定于项目的关注点不同这一问题。

在线服务提供商

路线图示例



在线服务提供商 (Online Services Provider, OSP) 从事构建 WEB 应用程序及其他网络可访问界面的核心业务。

由于最初目的是在不妨碍创新的前提下验证设计的整体稳固性，因而早期行动专注于体系结构审查和安全性测试。

由于重要的系统将面向网络，还应尽早执行基础架构强化措施，并逐渐解决托管环境所存在的风险。

尽管该措施基于组织的核心业务，但还是应该尽早启动标准与遵从措施，然后根据外部遵从驱动因素的关键性进行推进。

随着组织的逐渐成熟，可逐渐增加威胁建模、安全要求以及防御设计等措施，以便在确定基准安全预期后支持预先安全防御。

第 1 阶段

对员工进行技术安全意识的培训
EG 1 制定和维护技术指导制度
估计总体商业风险概况
SP 1 建立和维护保证方案
确定软件攻击面
AR 1 根据已知的安全要求分析设计
根据已知的安全要求，创建测试案例
ST 1 明确评估已知的安全测试案例
维护操作环境规范
OE 1 确定和安装关键的安全补丁

第 2 阶段

确定并监视外部标准遵从的驱动因素
SC 1 建立和维护标准遵从检查表
建立并维护每个应用程序的攻击树
TM 1 根据软件体系结构，制定攻击者配置文件
根据已知的安全要求，创建审查检查表
CR 1 对高风险代码执行定点审查
确定安全问题的联络点
VM 1 创建非正式的安全响应团队
捕获关于部署的重要安全信息
OE 1 记录典型应用程序警报的步骤

第 3 阶段

对员工进行针对特定角色的应用程序安全培训
EG 2 聘用安全指导员，增强项目团队
根据业务风险对数据和应用程序进行分类
SP 2 建立每个分类的安全目标
根据业务功能，制定安全要求
SR 1 使用指导原则、标准和标准遵从资源
检查是否完整提供了安全机制
AR 2 为项目团队部署设计审查服务
利用自动安全性测试工具
ST 2 将安全性测试集成到开发流程
建立基础架构补丁管理流程
IH 2 监视基准基础架构配置状态

第 4 阶段

创建应用程序安全支持门户网站
EG 3 建立基于角色的考试/认证制度
DD 1 维护推荐软件框架的列表
将安全原则显式应用于外围界面
AR 3 为敏感资源制定数据流图表
为体系结构审查建立发布关卡
CR 2 利用自动代码分析工具
将代码分析集成到开发流程
VM 2 建立一致的事件响应流程
采用安全问题披露流程
IH 3 确定和部署相关操作保护工具
扩展基础架构配置的审核程序

第 5 阶段

SC 2 建立安全和标准遵从的内部标准
建立项目审核实践
SP 3 定期进行业界范围的成本比较
收集历史安全费用的标准
TM 2 用应用程序特定的数据详细描述攻击树
采用一种衡量威胁的权重系统
SR 2 为每个项目建立和维护滥用案例模型
根据已知风险，指定安全要求
ST 3 采用应用程序特定的安全测试自动化
建立安全测试发布关卡
OE 2 创建每次发布的变更管理步骤
维护正式的操作安全指南

对组织的路线图有影响的 OSP 具有一些共同的特征。

外包式开发

对于采用外部开发资源的组织，通常需要限制代码的访问权限，这会导致组织优先执行安全要求任务，而不会先执行代码审查任务。此外，如果在较早的阶段执行威胁建模，组织便可以向外包开发人员提出更明确的安全要求。由于有关软件配置的专业技术通常是外包组中最重要的，因此必须制定合同以说明与操作实现相关的措施。

符合 PCI 标准的组织

对于必须满足支付卡行业数据安全标准 (Payment Card Industry Data Security Standard, PCI-DSS) 要求的组织，应该将标准与遵从措施放在路线图中的早期阶段。这使组织能够借机制定满足 PCI-DSS 标准要求的措施，还可以对以后的路线图进行相应的定制。

Web 服务平台

对于构建 Web 服务平台的组织，设计方面的错误将会带来其他风险，并且修复的成本更高。因此，应该将威胁建模、安全性要求和防御设计放在路线图的早期阶段。

通过收购而发展壮大的组织

在通过收购而发展壮大的组织中，通常存在遵循不同开发模式的若干个项目团队，其安全相关任务的等级也不同。对于这类组织，如果正在开发不同类型的软件，那么需要为各个部门或项目团队分别制定相应的路线图，以解决各个项目的起点和特定于项目的关注点不同这一问题。

案例研究

本节介绍了许多可供特定业务修改和使用的成熟度模型案例。将建议的路线图作为指导，这些案例研究将说明组织如何依据实际情况调整最佳做法，并将特定于组织的风险考虑在内。

VirtualWare

案例研究：中型 ISV

企业概况

VirtualWare 是其所在市场的领导者，主要提供集成的虚拟化应用程序平台，各个组织可以利用这些平台将各种应用程序界面整合到一个环境中。该公司将在为多种环境（包括 Microsoft、Apple 和 Linux 平台）构建的服务器应用程序和桌面客户端中提供该技术。

VirtualWare 是一家拥有 200 到 1000 名员工的中型企业，在大多数主要国家/地区都设有分支机构。

组织

Virtualware 从事核心软件平台的开发已 8 年多了。在这段时间内，由于该企业将 Web 界面的使用率降至最低，因此很少遭受常见 Web 漏洞带来的风险。大部分 Virtualware 平台均通过基于服务器的系统或在桌面上运行的胖客户端运行。

Virtualware 近期启动了许多新的项目流，可以通过 Web 技术提供客户端和服务器界面。正因为了解了 Web 上出现的常见攻击的范围，才促使该企业审查其软件安全战略，确保该战略能充分应对企业发展过程中可能遇到的各种威胁。

该企业以前对应用程序代码进行了基本的审查，当时企业更注重程序的绩效与质量，并不关注安全问题。Virtualware 开发人员使用许多代码质量分析工具来发现错误并在代码中解决这些错误。

基于这样的想法，高层管理团队设立了相应的战略目标，以审查应用程序安全性的当前状态，确定识别、清除和预防应用程序漏洞的最佳方法。

环境

VirtualWare 使用 Java、C++ 和 Microsoft .NET 等多种技术开发其虚拟化技术。其核心应用程序虚拟化技术是用 C++ 编写的，该技术已经过许多针对错误和安全性的审查，但目前还没有用来识别和修复已知或未知安全错误的正式流程。

尽管 Virtualware 的后端系统是使用 Microsoft 和 C++ 技术构建的，但该企业仍选择使用 Java 支持其 Web 技术。负责新 Web 界面的开发小组主要由 Java 开发人员组成。

Virtualware 拥有 300 多名开发人员，这些员工根据所从事的项目分成若干组。共有 12 个项目组，平均每组包含 20 至 40 名开发人员。每个项目组都很少遇到软件安全问题，虽然高级开发人员会对代码进行基本评估，但该企业并不将安全性视为关键目标。

Virtualware 的每个项目组均采用不同的开发模型。目前所采用的两种主要方法是灵活的 SCRUM 和迭代的瀑布型方法。几乎不需要 IT 部门或项目架构师提供有关软件安全性的指导。

重要挑战

- 迅速发布应用程序功能，以确保能够维持其领先的竞争优势
- 在软件安全性概念方面经验有限，目前在安全性相关的任务方面采取的措施非常少
- 企业内开发人员的新旧更替，老员工离职后，由缺乏经验的新员工代替
- 应用程序中采用了多种技术，其中一些旧应用程序自构建后从未进行过更新
- 不了解企业当前面临的安全状况或风险

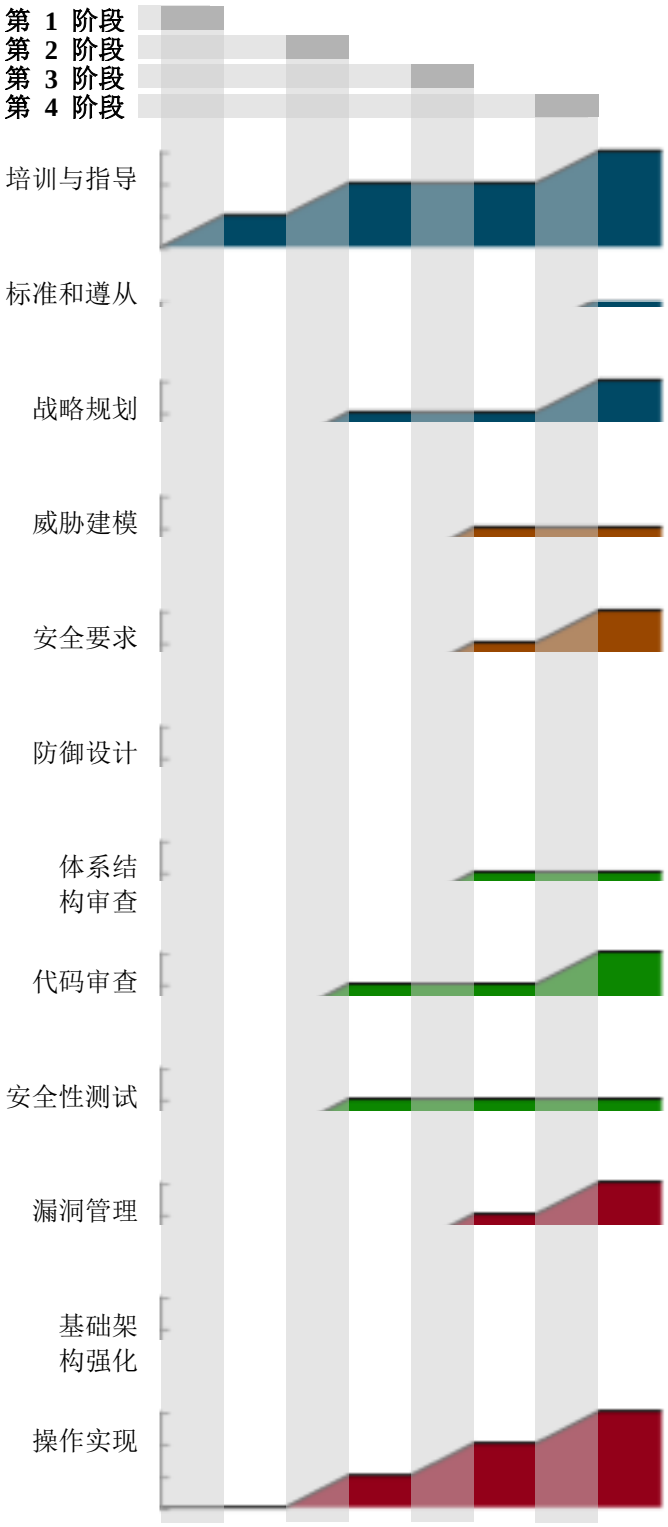
Virtualware 希望专注于确保采用安全的方式为客户提

供新的 Web 应用程序。因此，实现安全保证计划的最初关注点是对开发团队的培训，使其了解安全保证计划，并提供一些针对安全编码和测试标准的基本技术指导。

该企业以前通过 support@virtualware.net 地址接收错误请求和安全漏洞。然而，由于该地址是通用支持地址，因此不一定能将现有请求筛选到企业内相应的项目组并正确处理这些请求。Virtualware 已认识到实施正式的安全漏洞响应计划的必要性。

实施战略

在企业内采用安全保证计划是一项长期的战略，这项战略将深刻影响开发人员的开发理念，以及企业开发和交付业务应用程序的流程。采用这项策略需要一年多的时间，因为这种规模的企业在这段时间内实施此战略相对容易。



软件保证成熟度模型

VirtualWare — 第 1 阶段

第 1 阶段（从开始到第 3 个月）– 认识程度与规划

Virtualware 以前就认识到它对于应用程序安全威胁的了解和关注非常有限，并且在安全编码方面也缺乏经验。在 Virtualware 内部署的第一阶段主要是对开发人员进行培训，实施指导和计划以确定当前的安全漏洞。

Virtualware 内部的开发团队在安全编码技巧方面缺乏经验，因此，Virtualware 制定了初期培训计划，为企业内的开发人员提供关于防御性编程技巧的培训。

由于企业拥有 300 多名开发人员且支持多种语言，Virtualware 面临的一项重要挑战是为开发人员提供一项在技术上足以使他们理解一些安全编码基本概念的培训计划。初期培训课程的主要目标是介绍编码技巧和测试工具。该课程在企业内部开展和教授，为期一天，涵盖安全编码的基本内容。

Virtualware 认识到其开发的许多应用程序都存在安全漏洞，但目前没有实际的战略能够识别出存在的漏洞并在合理的时间范围内处理这些风险。采用基本的风险评估方法后，该企业便可以对现有的应用程序平台进行审查。

该阶段还包括贯彻许多安全理念，以便开发团队强化安全工具。开发团队目前已有很多可用于执行质量类型评估的工具。并且对代码审查和安全测试工具进行了其他调查。

SAMM 目标

在项目的这一阶段，Virtualware 实现了以下 SAMM 功能和目标。

措施

EG 1
SP 1
SR 1
CR 1
ST 1
VM 1

为了实现上述成熟度级别，Virtualware 在开展这一阶段时实施了多项计划。采用了以下行动方案；

- 为所有开发人员提供了为期一天的安全编码课程（高级）培训；
- 制定了一份针对企业内所用技术的应用程序安全性的技术指导白皮书；
- 创建风险流程，为应用程序平台执行高级业务风险评估，并审查业务风险；
- 为开发人员准备初步的技术指导原则和标准；
- 对可能对企业造成巨大风险的应用程序平台执行快速代码审查；
- 为项目开发测试案例和用例，并针对应用程序评估案例；
- 为应用程序安全措施指定角色；
- 为保证计划的下一阶段生成战略路线图草案。

由于 Virtualware 企业本身拥有的专业技能有限，该公司与第三方安全咨询小组合作，希望其协助完成培训计划的创建过程，并为该公司制定威胁模型和战略路线图。

这一阶段所面临的重要挑战之一是让所有 300 名开发人员学完为期一天的培训课程。要实现这一目的，Virtualware 需要安排 20 天的课程培训，只需要每组的少数开发人员同时参加该课程。这样，可以减少培训期间对员工资源的整体影响。

在项目的这一阶段，Virtualware 在采用风险审查过程以及审查企业的业务风险方面投入了大量资源。尽管企业在这些任务上花费了相当大的精力，但为了确保 Virtualware 实施的下一步骤可应对企业面临的业务风险，在这方面花费的精力必不可少。

Virtualware 管理层收到了企业中大部分开发人员对于培训计划的积极反馈。尽管培训课程并不详尽，但开发人员仍认为初期培训为其提供了能够立刻应用到日常安全编码工作中的基本技巧。

实施成本

在项目的这一阶段中，投入了大量的内部资源和成本。本阶段涉及以下三种不同类型的成本。

内部资源要求

在本阶段中，创建内容、举办研讨会以及审查应用程序安全措施需要占用内部资源。工作量按照每个角色需要的总天数计算。

培训资源要求（培训期间每个人需要参加的培训天数）

Virtualware 的每个开发人员都需要参加培训课程，因此每个开发人员都必须用一天的时间学习应用程序安全计划。

外包资源

由于 Virtualware 缺乏相关知识，在创建内容以及为开发人员创建/提供培训计划时需要利用外部资源。

开发人员	14 天	企业所有者	8 天
架构师	10 天	QA 测试人员	3 天
管理人员	8 天	安全审核员	9 天

开发人员（每人）	1 天

咨询人员（安全性）	15 天	咨询人员（培训）	22 天

VirtualWare — 第 2 阶段

第 2 阶段（从第 3 个月到第 6 个月）– 培训与测试

Virtualware 在第一阶段认识到许多应用程序中包含容易被外部威胁利用的漏洞。因此，本阶段的重要目的之一是实施基本的测试和审查功能，以确定漏洞并在代码中处理漏洞。

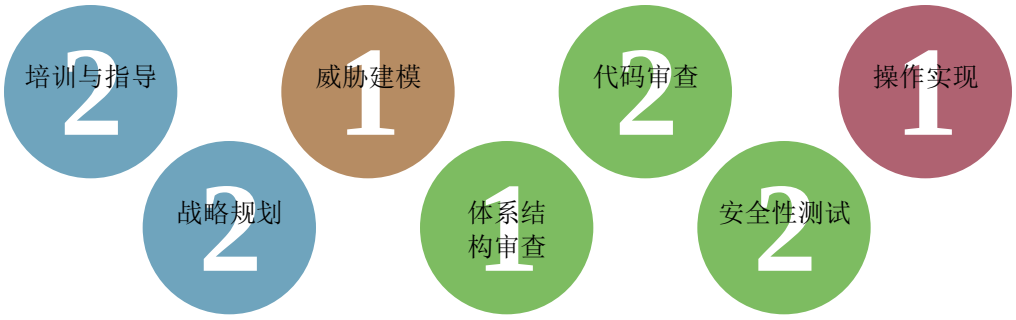
本阶段引入了自动化工具，使用该工具可以确定代码覆盖率，找出程序中的缺陷。这是本阶段实施过程中面临的最大挑战。传统上，开发人员以前使用自动化工具时遇到巨大困难，因此实施新工具被视作一项巨大的挑战。

为了确保在企业内成功利用自动化工具，Virtualware 采用了试推广策略。应先将该工具分配给小组高级负责人，其他开发人员在一段时间内逐渐采用这些工具。公司鼓励各小组采用该工具，但尚未确定可利用的正式流程。

实施这一阶段还会引入更正式的培训和认识计划。参加过上次培训的开发人员需要参加 Web 服务和数据验证领域的更为具体的培训。这个为期 6 小时的具体的新培训课程就是针对这两个专门领域开发的。Virtualware 还会对企业架构师、管理人员实施其他培训，使整个企业都对安全性予以关注。

SAMM 目标

在项目的这一阶段，Virtualware 实现了以下 SAMM 功能和目标。



措施

EG 2
SP 2
TM 1
AR 1
CR 2
ST 2
OE 1

为了实现上述成熟度级别，Virtualware 在开展这一阶段时实施了多项计划。采用了以下行动方案；

- 为 QA 测试人员、管理人员以及架构师提供其他培训课程；
- 对数据资产进行分类并设置安全目标；
- 将风险评估方法开发成包含攻击树和配置文件的威胁建模方法；
- 审查并确定每个应用程序平台的安全要求；
- 引入自动化工具，以便为现有应用程序和新代码库确定代码覆盖率和安全分析；
- 审查并增强现有的渗透测试计划；
- 增强现有的 SDL，以便将安全测试作为开发流程的组成部分加以支持

Virtualware 对现有的应用程序安全培训计划进行了修改，使其成为业务应用程序安全认识计划的一个技术性不很强的小型版本。该计划将课程时间缩短为 4 小时，可以参与该课程的人员扩展到企业的管理人员和企业所有者。

针对现有代码审查和渗透测试计划的高级审查表明该流程不完善，还需要进一步增强该流程，以便针对应用程序安全漏洞提供更好的测试和结果。小组开始实施用于执行渗透测试和代码审查的新计划。作为该计划的一部分，应为项目组中的每位高级开发人员分配 4 天左右的时间，对其应用程序执行高级源代码审查。

Virtualware 管理层知道基础架构与应用程序是紧密集成的，在这一阶段中将会审查应用程序平台（基础架构）的操作端。本阶段会查看建议部署的硬件与应用程序接口之间的基础架构要求和应用程序整合功能。

在本阶段中，项目组会审查针对应用程序安全性的战略路线图和方法。此次审查和更新的目标是正式对数据资产进行分类，并设置与数据资产和应用程序相关联的相应业务风险级别。随后，项目组便可以为这些应用程序设置安全目标。

实施成本

在项目的这一阶段中，投入了大量的内部资源和成本。本阶段涉及以下三种不同类型的成本。

内部资源要求

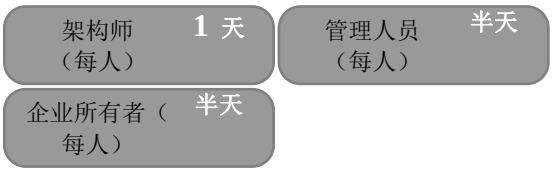
在本阶段中，创建内容、举办研讨会以及审查应用程序安全措施需要占用内部资源。工作量按照每个角色需要的总天数计算。

培训资源要求（培训期间每个人需要参加的培训天数）

Virtualware 的每个开发人员都需要参加培训课程，因此每个开发人员都必须用一天的时间学习应用程序安全计划。

外包资源

由于 Virtualware 缺乏相关知识，在创建内容以及为开发人员创建/提供培训计划时需要利用外部资源。



VirtualWare — 第 3 阶段

第 3 阶段（从第 6 个月到第 9 个月）– 体系结构与基础架构

要在 Virtualware 内实施保证计划的第三阶段，必须在之前实施的各阶段的基础上实施，注重风险建模、体系结构、基础架构以及操作实现功能。

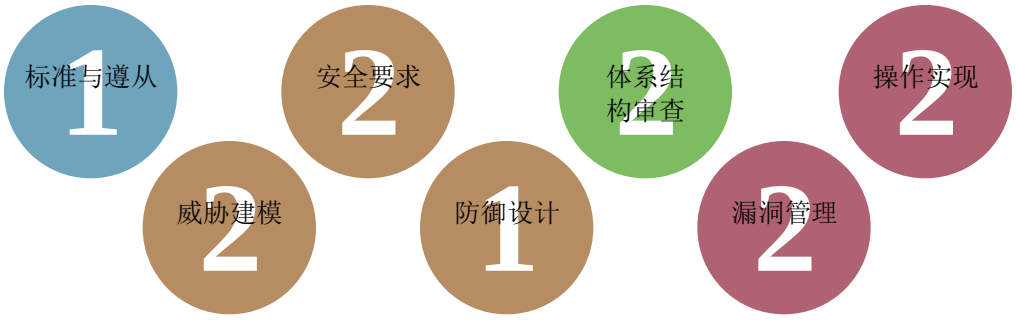
本阶段面临的重要挑战是在应用程序平台与企业的操作端之间进行更为紧密的整合。在以前的阶段中，已经为 Virtualware 项目组介绍了漏洞管理和应用程序安全性的操作端。在本阶段中，Virtualware 采用了关于这些方面的下一阶段操作，并引入了清晰的已处理的事件响应，并详细修改了控制流程。

Virtualware 选择在这个实施过程中启动两个新领域。虽然 Virtualware 不受法规遵从的影响，但该企业的客户开始询问这些平台能否帮助通过法规遵从要求。Virtualware 已经成立了专门小组，负责找出相关的标准遵从驱动因素并创建驱动因素检查表。

在以前的阶段中，Virtualware 引入了许多新的自动化工具，这些工具可用于审查和识别漏洞。虽然这些工具并不是本阶段的重点，但开发小组采用了这些新工具，并报告在项目组中采用这些工具使他们受益匪浅。

SAMM 目标

在项目的这一阶段，Virtualware 实现了以下 SAMM 功能和目标。



措施

SC 1
TM 1
SR 2
DD 1
AR 2
VM 2
OE 2

为了实现上述成熟度级别，Virtualware 在开展这一阶段时实施了多项计划。采用了以下行动方案；

- 为企业内的项目定义和发布有关安全要求及防御设计的技术指导；
- 明确和记录遵从性和法规要求；
- 明确并创建应用程序基础架构的安全准则；
- 创建经认可的开发框架的定义列表；
- 改进 Virtualware 目前所采用的威胁建模流程；
- 采用事件响应计划并准备安全问题披露流程；
- 引入适用于所有项目的变更管理流程及正式的指导原则。

为了与先前阶段为开发人员引入自动化工具保持一致，还为企业引入了有关安全编码技巧的正式技术指导。这些指导是与语言和技术相关的具体技术文档，提供了关于每个相关语言/应用程序安全编码技巧的指导。

借助培训和认识计划中的综合方法、技术指导信息，然后引入对开发人员有所帮助的自动化工具，Virtualware 开始发现即将提供给投入生产的应用程序的代码有了明显的不同。开发人员对该计划为其提供的工具和培训给出了积极肯定的反馈。

Virtualware 的项目组首次负责应用程序平台的安全和设计工作。在本阶段中，每个项目组会执行正式的审查流程并针对最佳做法进行验证。部分项目组发现待审查的安全和业务设计之间存在漏洞。建立正式的计划以确保消除了这些漏洞。

在项目的这一阶段中引入了正式的事件响应计划及变更管理流程。这是一个难以实施的流程，Virtualware 项目组最初就是因为该流程对企业文化和操作端的影响非常大，在实施时非常费力。不过，随着时间的推移，每个小组成员都认识到新流程的价值，并在实施期间接受了该流程所带来的变化。

实施成本

在项目的这一阶段中，投入了大量的内部资源和成本。本阶段涉及以下三种不同类型的成本。

内部资源要求

在本阶段中，创建内容、举办研讨会以及审查应用程序安全措施需要占用内部资源。工作量按照每个角色需要的总天数计算。

外包资源

由于 Virtualware 缺乏相关知识，在创建内容、为项目组创建/提供流程、指导原则以及为项目组提供帮助时需要利用外部资源。

VirtualWare — 第 4 阶段

第 4 阶段（从第 9 个月到第 12 个月）– 管理与操作安全性

要在 Virtualware 内实施保证计划的第四阶段，也必须以之前实施的各个阶段为基础，增强企业现有的安全功能。到现在为止，Virtualware 已实施了许多重要的应用程序安全流程和机制，确保能够安全地开发和维护应用程序。

本阶段的重中之重是支持调整与管理规范。这三种功能对于创建长期有效的应用程序安全战略具有重要作用。实施了完整的培训计划之后，也应该同时为 Virtualware 制定长期的战略路线图。

本阶段的另一个重点在于实施的操作端。Virtualware 管理层以前就认识到，事件响应计划和专用的变更管理流程对于长期战略至关重要。

Virtualware 将这一阶段视作长远发展的跳板。在这一阶段，企业实施了许多最终措施，以便使之前各阶段进行的现有安全工作贯穿成一个整体。从长远来看，这样可以确保实施的过程、理念和控制对于企业长期有效，企业生产的应用程序平台也最安全。

Virtualware 选择在此阶段为客户引入新的应用程序安全计划，为 Virtualware 客户详细介绍有关应用程序安全性、安全部署应用程序以及报告 Virtualware 应用程序中的漏洞的一系列计划。这些计划的重要目标是使客户群树立信心，让他们相信 Virtualware 应用程序在构建时考虑了安全性。Virtualware 还能帮客户确保采用其技术的应用程序环境的安全性。

SAMM 目标

在项目的这一阶段，Virtualware 实现了以下 SAMM 功能和目标。

措施

EG 3
SC 2
SP 3
SR 3
CR 3
VM 3
OE 3

为了实现上述成熟度级别，Virtualware 在开展这一阶段时实施了多项计划。采用了以下行动方案；

- 为所有项目创建定义完整的安全性要求和测试计划；
- 创建并实施事件响应计划；
- 审查了应用程序现有的警报程序，记录捕获事件的流程；
- 创建安全部署应用程序的客户安全性白皮书；
- 审查项目内现有的安全开支，确定是否为每个项目的安全性分配了适度的预算；
- 针对应用程序角色，实施最终培训和关注计划；
- 为企业制定完整的长期应用程序安全性战略路线图。

在先前的阶段中，Virtualware 已经发布了正式的事件响应计划，使客户可以提交其代码中的漏洞。在这一阶段中，Virtualware 会利用所提交漏洞的结果，对问题出现的原因和方式进行评估，并尝试进行一系列报告，以确定在已报告的漏洞中识别的一般主题。

作为确保应用程序在企业内部和客户网络上安全部署的持续努力的一部分，Virtualware 创建了一系列白皮书，在业界标准的基础上为客户提供建议的基础架构强化服务。这些方针的目的是帮助客户选择最佳的方法部署其应用程序。

Virtualware 在此阶段实施了基于 CBT 的短期培训，使现有的和新入职的开发人员能够保持应用程序安全方面的技巧。该企业还强制要求所有与“应用程序”相关联的角色每年都必须接受为期一天的培训课程。这是为了确保对开发人员教授的技巧不会被丢掉，而且使新的开发人员能够在公司工作期间在技术方面有所进步。

在 Virtualware 内实施的最后一个功能是完成一个“按原样”漏洞评估和审查，确保过去一年的工作取得了哪些效果。进行这一简短的计划期间，企业向所有参与的团队成员发放调查问卷，并对 SAMM 进行基准审查。在该审查过程中确定的缺陷和优势将记录在企业的最终战略路线图中，用来制定 Virtualware 的下一年度战略。

实施成本

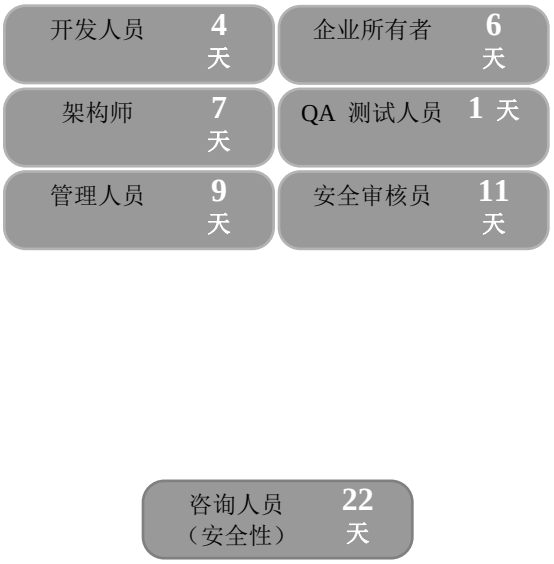
在项目的这一阶段中，投入了大量的内部资源和成本。本阶段涉及以下三种不同类型的成本。

内部资源要求

在本阶段中，创建内容、举办研讨会以及审查应用程序安全措施需要占用内部资源。工作量按照每个角色需要的总天数计算。

外包资源

由于 Virtualware 缺乏相关知识，在实现该阶段（包括文档、流程和研讨会）时需要利用外部资源。



后续阶段（从第 12 个月再往后）

过去 12 个月中，Virtualware 实施了许多培训和教育计划，以此开始制定内部的准则和政策。在保证计划实施的最后阶段，Virtualware 开始在内部进行发布，并与客户协作，增强其客户应用程序平台的安全性。

Virtualware 管理层确立了根本使命：确保企业内开发的软件是安全的；确保市场注意到他们采取的安全行动；帮助用户保护他们的应用程序平台。

为了达到这些管理目标，前 12 个月为在 Virtualware 内有效实施战略铺平了道路，Virtualware 最终将帮助客户保护他们的应用程序环境。在发展过程中，Virtualware 在企业内部开展了许多活动，以确保该企业不会“犯老毛病”。其中一些活动包括：

- 企业所有者和团队领导注意到与其应用程序相关的风险，在应用程序发布前他们需要签字；
- 团队领导现在要求所有应用程序都要正式接受安全检查，开发人员每周会进行一次代码审查；
- 持续对所有项目员工每年开展培训和教育计划（包括 CBT），要求开发人员每年至少参加一次课程培训；
- 设立了专门负责应用程序安全性的团队领导，该职位负责客户交流、客户技术文件和安全准则。

随着计划的推进，Virtualware 现在拥有了将安全纳入其 SDL 的文化，确保为客户开发和提供的应用程序安全可靠。一个有效的流程已经建立，漏洞可以报告给企业并在需要的时候由企业进行处理。

在最后的实施阶段，需要执行项目漏洞评估，以确定实施期间出现的任何缺陷。特别是，由于员工流动性大，Virtualware 需要不断地在新开发人员入职时对他们进行培训。为处理这一问题而设定的一个关键目标是专为开发人员引入入门指导计划，使他们能够在入职时接受正规的安全培训。这也会帮助树立安全在企业及其开发团队中非常重要的理念。

成熟度记分卡

成熟度记分卡是 Virtualware 在实施软件保证计划过程中作为自我评估完成的。最终的记分卡（如右图所示）表示 Virtualware 在此次实施的第四阶段结束时的状态。